



Título del Proyecto:

Juego de preguntas onLine (Trivia_UPV)

Autor:

Parra Hernández
Miguel Ángel

Director:

Tomas Girones Jesús

**TESINA PARA LA
OBTENCIÓN DEL TÍTULO DE:**

**Máster en Desarrollo de
Aplicaciones sobre Dispositivos
Móviles**

Septiembre del 2016

Contenido

Introducción	3
Descripción del problema	3
Objetivos	3
Motivación	3
Situación de... / Tecnologías utilizadas.....	3
Arquitectura de la aplicación	5
Esquema del diseño	5
Modelo de datos	7
Vistas	9
Conclusiones	11
Anexo fuentes	12
Listado de fuentes entregadas o enlace a GitHub	12

Introducción

Descripción del problema

El proyecto surge por un lado por sugerencia de la dirección del master y por otro lado de la experiencia, en el transcurso del mismo observando el desarrollo logrado por el profesorado de la UPV en la evaluación en línea de estudiantes de nuevas tecnologías, un campo en permanente crecimiento y renovación, donde el perfil de la educación incorpora progresivamente a entidades académicas de menor tamaño que emplean la tecnología para mejorar el desempeño académico, ahorrar costes y mejorar su competitividad.

Objetivos

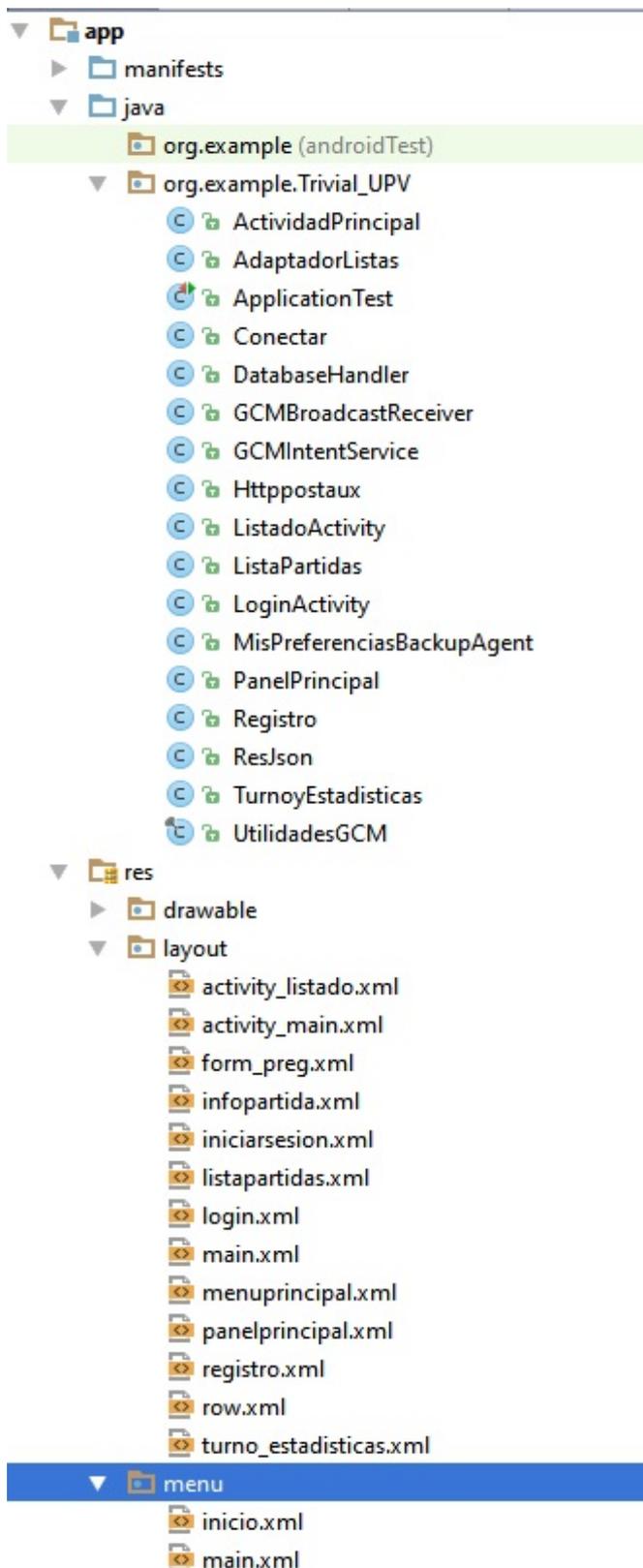
Lograr una aplicación para estudiantes con enfoque educativo, auto-evaluativa, en línea que ofertará a los alumnos repaso de materias (en especial Diplomado de Android), en el cual el estudiante podrá resolver test de prueba con respecto a una materia determinada, sin estar en la biblioteca o frente a un pc, el formato de la aplicación es similar al usado para los test de estudiantes en línea y que en nuestro caso permite interacción con otros usuarios en forma de juego trivia con sus respectivos puntajes sin perder de vista esto, que es un juego.

Motivación

La motivación personal fue profundizar en los conceptos de interacción entre diferentes tecnologías web y Android.

Situación de... / Tecnologías utilizadas

El esquema general utilizado es el almacenamiento en servidor y mensajería en la nube. Servidor web utilizando Apache, PHP y base de datos MySql el código PHP y el diseño de la BD fue proporcionado en gran parte por el director del proyecto Con algunas modificaciones, complementamos el desarrollo de la aplicación



Como funciones del código php se encuentra gestionar las peticiones del cliente, escribir, actualizar o borrar registros de la BD y enviar de vuelta el cambio de turno del jugador.

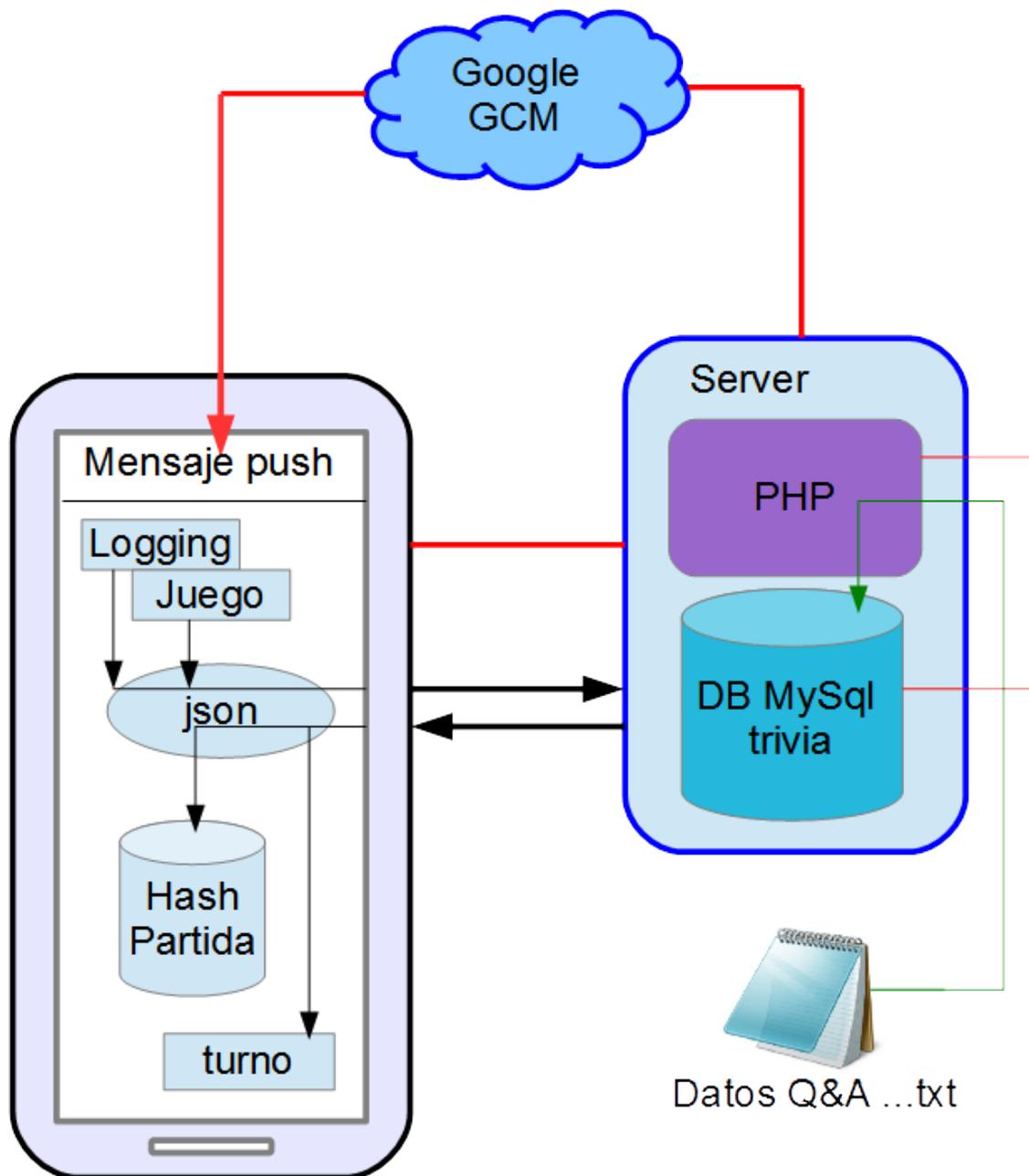
Interacción-comunicación con el cliente y el server mediante tecnología REST, junto con notificaciones PUSH con el cliente Android, para avisar del turno, también para inicializar el chat de recordatorio o invitación a jugar.

Como podemos apreciar en la figura al lado (Figura 1) la aplicación cliente Android usa Json para codificar y decodificar los datos arrojados en la comunicación con el Server, en el cliente se usa asyntask para evitar el bloqueo de la UI, también listView, cuadros de dialogo, etc.

Las notificaciones en la nube usan el GCMIntenService.java junto con GCMBroadcastReceiver.

Arquitectura de la aplicación

Esquema del diseño



Con el fin de jugar, logeamos (Vista 1) al usuario con el nombre o Nick y contraseña, el correo electrónico y la identificación GCM del móvil son usadas internamente para identificación y envíos de mensajes push con estas cuatro características es posible



tener a dos o más jugadores (estudiantes) compartiendo el móvil sin que las calificaciones sean confundidas, ya que el mensaje llegara identificando al destinatario.

Cumplido el paso anterior entramos a una listView personalizado donde encontramos los temas o juegos a los que se encuentra inscrito el jugador junto con la información de cuantos jugadores tiene la partida y si tiene el turno o no (Vista 2). Si tiene el turno y con clic sobre el tema contesta la pregunta escogiendo de las cuatro o cinco opciones que se le presentan, seguido a esto aparece un dialogo con el resultado es decir si acertó o no y los puntos ganados, al aceptar con el botón regresa a la vista de los temas, donde ya aparecerá sin turno.

En esta vista también tiene la posibilidad con los botones inferiores de entrar o salir de una partida, para esto aparece un cuadro de dialogo con la información de la respectiva partida, cuantos jugadores admite o si ya no hay cupo para entrar. Si hay cupo y acepta entra a jugar en ese tema y aparecerá en la lista la nueva partida.

El trinomio Apache PHP y MySql de utilización muy extendida en los hostings comerciales fue escogido por esto y también es una tecnologia relativamente sencilla y está disponible como software libre.

Comparado con SOAP, REST tiene la ventaja de apoyarse en el protocolo HTTP, necesita menos recursos tanto de código como de carga en la trasmisión de datos, y por otro lado SOAP no tienen respaldo en el SDK de Android, de las librerías necesarias.

La base de datos se alimentan mediante un script php dentro del server (http://localhost/trivia/Leer_txt.php) que transforma los archivos .txt con el formato de la UPV usados en test de repaso en el Diploma de Android y los convierte en un array de bloques en los que se encuentran preguntas, respuestas y numero de respuesta correcta, siguiendo este formato se pueden insertar los ficheros que se quieran hasta el máximo soportado por MySql o el Hosting en su defecto.



La tabla reglas contiene el número de jugadores mínimo para iniciar una partida y el máximo que pueden jugar aquí, junto con su nombre

Otra tabla muy importante es la de pregunta que contiene los temas las preguntas, las respuestas, el número de respuesta correcta y algunas estadísticas como cuantas veces una pregunta ha sido acertada por los jugadores.

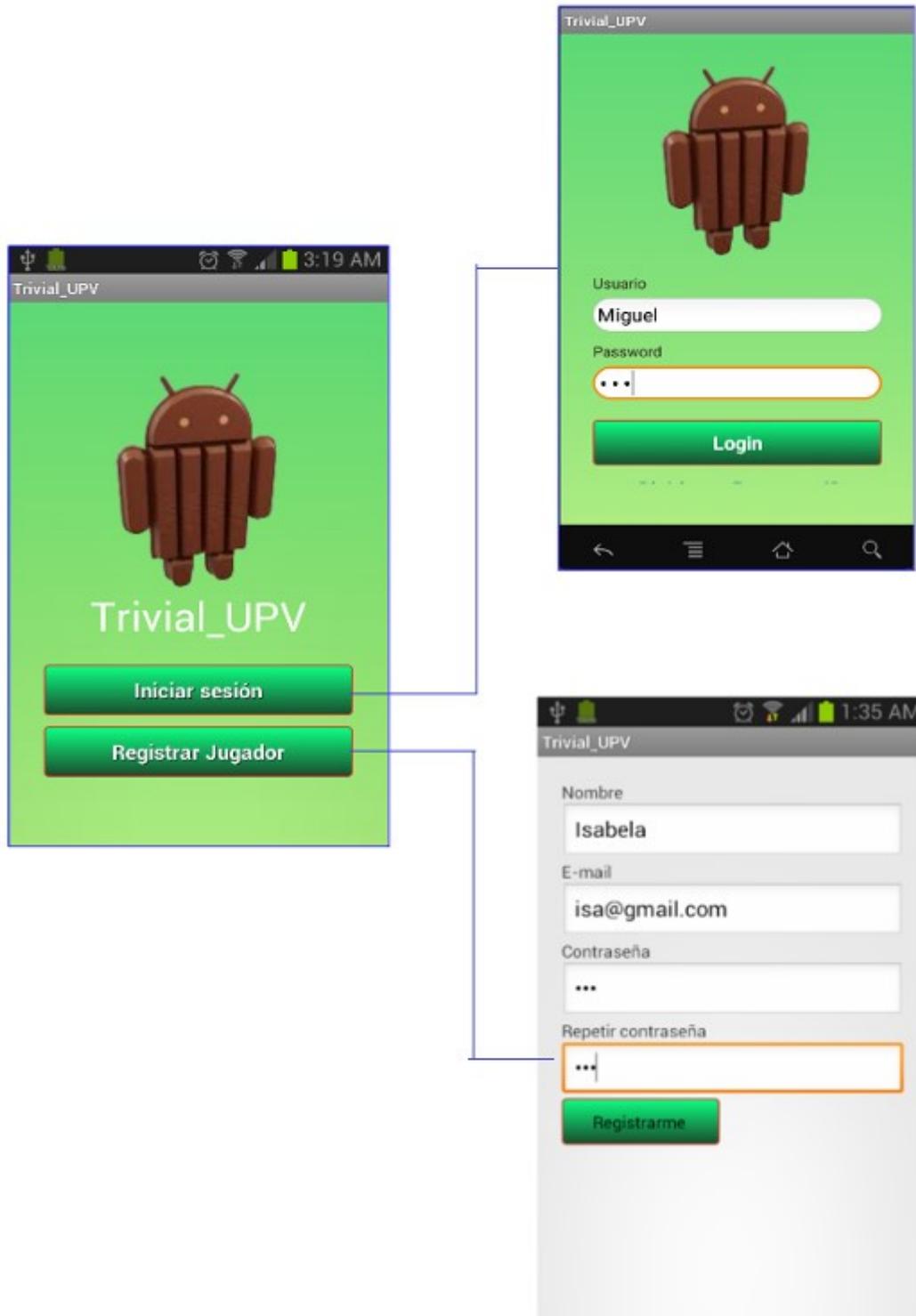
De acuerdo al resultado que se obtenga con las peticiones del cliente sobre la BD el código php devolverá un código numérico en la mayoría de los casos este response es transmitido a el cliente que enrutara el código de la aplicación a las acciones pertinentes, por ejemplo si una respuesta es correcta devuelve "0" y en la actividad turnos tenemos:

```
if (arrayRes[0].equals("0")) {  
    builder.setTitle("CORRECTA");  
    builder.setMessage("10 PUNTOS GANADOS");  
    AlertDialog dialog = builder.create();  
    dialog.setButton("Aceptar", new  
DialogInterface.OnClickListener()
```

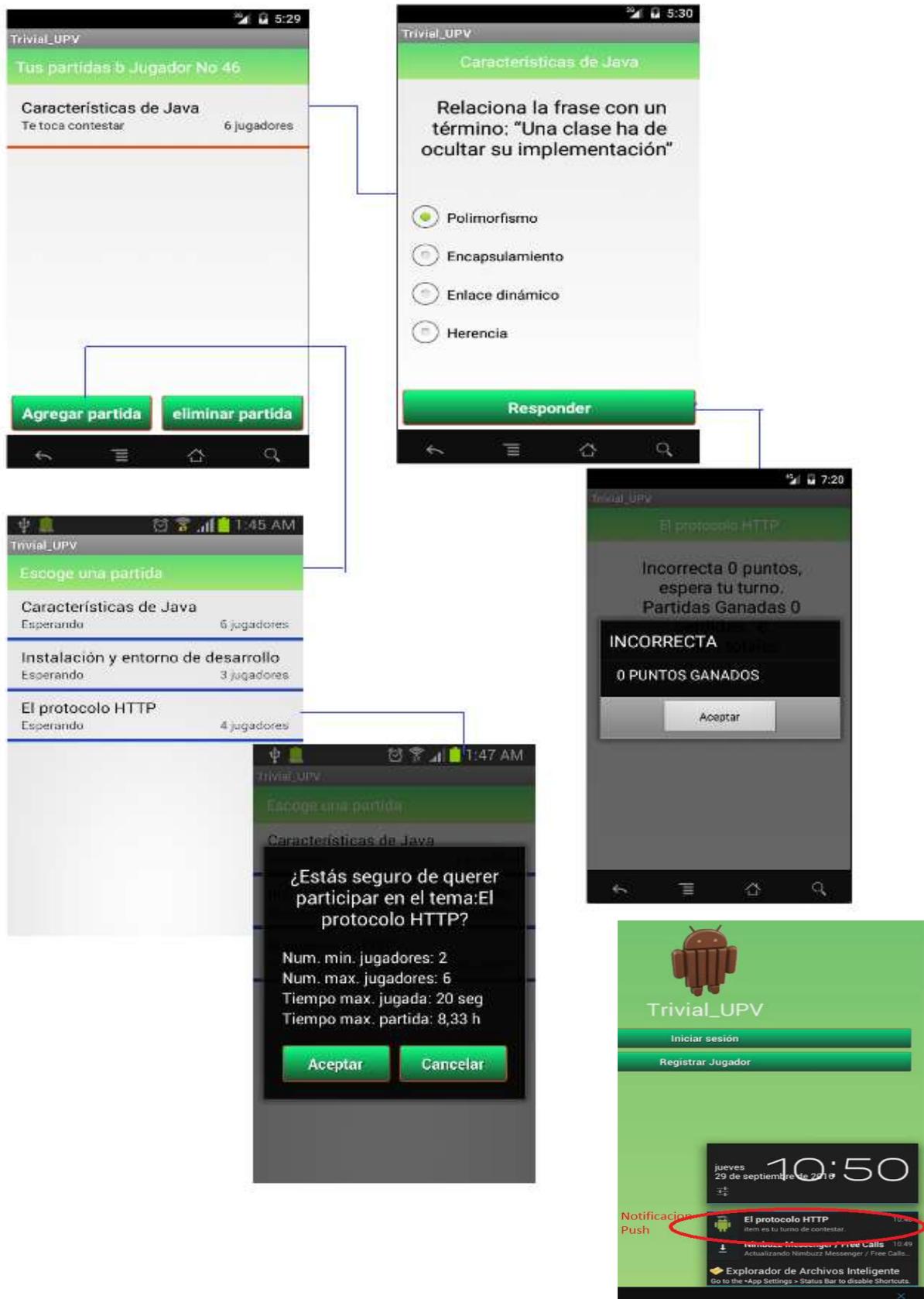
lo que llamará el cuadro de dialogo para presentar el resultado "CORRECTA"....



Vistas



Vista 1. Relaciones entre las vistas de Login y registro



Vista 2. Relaciones entre las vistas del juego

Capítulos adicionales

Conclusiones

Los objetivos se cumplieron en su mayoría ya que es un juego multiusuario que se puede usar con conexión a internet (online). Aun cuando hay problemas con las notificaciones PUSH que en este momento la plataforma no está registrando todos los móviles a los que se les aplica el procedimiento, todavía se puede jugar entre los que si se dejan registrar . Se intentó varias veces actualizar a Firebase pero se acaballaron varios errores entre ellos la actualización a Android Studio 2.2 que genera problemas con una característica nueva en la que interviene Glade y que tiene que ver con instalación al vuelo (Cualquier modificación se instala dentro de un proceso de instalación es decir los cambios en el código se reflejan casi instantáneamente en la aplicación sin reinstalarla).

Objetivos puntuales que no se cumplieron:

La aplicación esta tomando una sola pregunta de cada tema en la base de datos, cuando debía rotarlas hasta agotar el tema.

La actualización de estadísticas no aparecen después del dialogo de respuesta correcta o incorrecta.

Falto el chat en el que un jugador podía comunicarse con otro para invitarlo a entrar al juego o como recordatorio para que juegue, sin embargo el código ya se encuentra adaptado y el método se llama msjOnline en ActividadPrincipal.java

Líneas abiertas.

Pasar los gráficos (o datos) de MySql con estadística diaria desde el servidor al cliente



Anexos

Listado de fuentes entregadas

Archivo Zip con:

APK de la aplicación	(Trivial_UPV_ParraM.apk)
Carpeta con el código fuente Cliente	(Trivial), elimine carpeta app/build
Carpeta código fuente server	(triviaServer)
Carpeta archivos MySql	(MySql)
Datos Tester	trivial (s).sql
DB limpia	trivial (7).sql
Memoria	(Trivia_UPV.pdf)

Se probó con servidor en local con "<http://192.168.0.3:80/trivia/>"