

LABORATORIO 5 – CASO DE ESTUDIO RECETAS EFECTIVAS

DESCRIPCIÓN

Recetas Efectivas (RE) ofrece ahora la posibilidad de interactuar con los ingredientes de una de las recetas que se va a preparar desde un dispositivo wearable tipo reloj.

En el laboratorio 5 el trabajo se centra en la interacción básica con información en el dispositivo wearable y comunicación con el teléfono.

La aplicación sobre la cual se desarrolla el laboratorio, debe ofrecer las siguientes funcionalidades:

Ver ingredientes de receta: Obtener los ingredientes (y sus características) de una receta predefinida para desplegarlos en el reloj. El reloj debe mostrar uno a uno los ingredientes que aún no han sido comprados para que el usuario indique de manera intuitiva si lo adquirió o no (en el supermercado, por ejemplo).

Ver datos básicos de ingrediente: Mostrar características de ingredientes de una receta predefinida en la interfaz gráfica de un dispositivo Wearable. La información mostrada al usuario corresponde al nombre y cantidad del ingrediente.

Registrar compra de ingrediente: Permite registrar una compra de ingrediente (previamente entregado por la aplicación móvil) desde una interfaz tipo wearable. Esta acción debe reflejarse en el modelo de datos de la aplicación en el teléfono.

Registrar ingrediente pendiente: El usuario debe poder marcar un ingrediente como “Pendiente de compra” utilizando una interfaz wearable. Esta acción debe reflejarse en el modelo de datos de la aplicación en el teléfono.

REQUERIMIENTOS FUNCIONALES

En este laboratorio se desarrollan los siguientes requerimientos funcionales sobre el dispositivo reloj:

- RF 1. Ver ingredientes de receta
- RF 2. Ver datos básicos de ingrediente
- RF 3. Registrar compra de ingrediente.
- RF 4. Dejar ingrediente como pendiente de compra

RESTRICCIONES DE LA APLICACIÓN

La aplicación debe aprovechar las capacidades gráficas y de interacción ofrecidas por el API de Android y Android Wear.

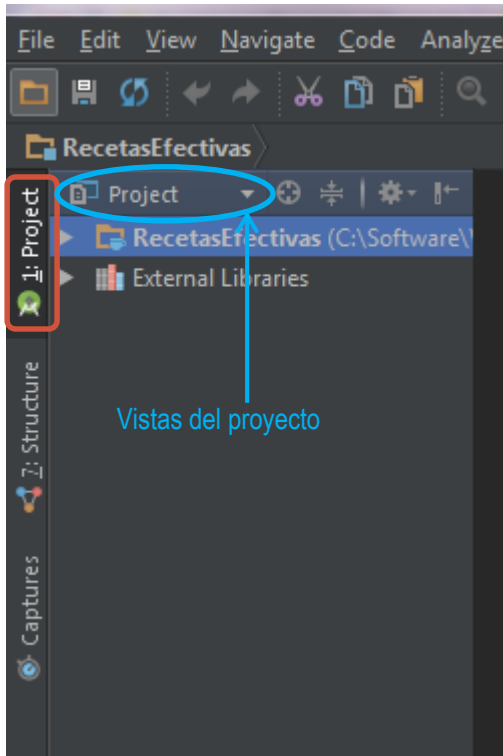
La información que intercambian ambas plataformas debe estar soportada por los APIs que provee Google. Revise los enlaces a la documentación en línea que se encuentran en el documento de Enunciado.

LO QUE USTED DEBE HACER

Los elementos subrayados en **color rojo** se deben entregar en el documento de respuestas junto con los demás entregables del laboratorio. Recuerde **Insertar el comentario ToReview en cada modificación que haga al código**, estos serán los únicos cambios revisados.

1. Descargue el esqueleto del laboratorio. Siga los siguientes pasos para abrirlo:
 - Abra Android Studio
 - En el menú de bienvenida de Android Studio seleccione “Open an existing Android Studio project”
 - Ingrese la ruta en donde descargó el proyecto.
 - Click en Ok

Si inicialmente no observa nada, abra la pestaña “1:Project” que se encuentra a la izquierda en la ventana de Android Studio, como se muestra en rojo en la siguiente imagen:



Esta pestaña le permite observar los archivos que componen el proyecto bajo diferentes vistas. Debe asegurarse que esté seleccionada la vista “Project” para la realización de los siguientes incisos, sin embargo, se sugiere investigar la vista Android, la cual da una estructura de proyecto más simple que permite tener un acceso rápido a los recursos más importantes de una aplicación Android.

Estructura de Proyectos

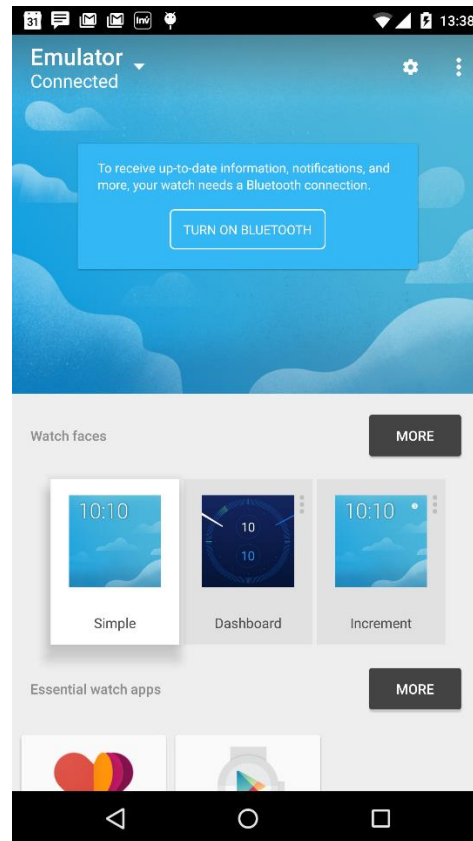
Revise los diferentes archivos y carpetas que componen al proyecto “Recetas Efectivas”. Note que en esta ocasión cuenta con dos módulos o partes de la aplicación: **mobile** y **wear**. Es importante que comprenda que en el ambiente Android Studio un proyecto es una solución software completa, que puede estar compuesta de diferentes módulos, los cuales pueden ser aplicaciones móviles.

2. Revise los archivos ANDROID MANIFEST de **ambos módulos e indique cuáles elementos del API de Android Wear observa y qué función cumplen**. En particular revise aquellos que participan en la comunicación entre plataformas.
3. Revise en ambos módulos cuáles son las clases Java encargadas de dicha comunicación. Revisando el código entienda cuál es el mecanismo de intercambio de información. Apóyese en la documentación oficial.

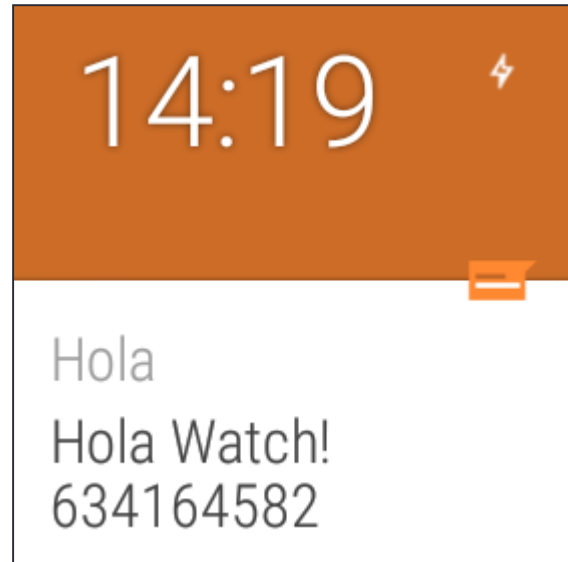
Despliegue en emulador

4. En Android Studio haga click en AVD Manager para crear un emulador de reloj. Asegúrese de configurarlo con arquitectura X86 (virtualización de procesador) y Android Lollipop (API 22).

5. Inicie el emulador haciendo click en el botón PLAY. Una vez inicie el sistema operativo salte el tutorial de arranque que ofrece Android.
6. Revise el enlace <http://developer.android.com/training/wearables/apps/creating.html> y siga las instrucciones para emparejar su teléfono con el emulador (a partir del punto 4).
7. Asegúrese de utilizar la aplicación Android Wear (descargable desde Play Store) en su teléfono para el emparejamiento. Antes de continuar confirme que el estado del emulador sea **Conectado**.



8. Realice una prueba de intercambio de mensajes entre su dispositivo y el emulador. Para ello despliegue el módulo "mobile" en su teléfono y haga click sobre el botón inferior. Revise la notificación recibida en el emulador y explique el bloque de código en el módulo **mobile** que logra que dicha notificación sea enviada.



Manejo de eventos

9. Inicie la aplicación desde el emulador del reloj. Haga click en el botón **Empecemos** y escriba con ayuda del teclado el comando de voz “comprar ahora” (simulando el micrófono del dispositivo).
10. La acción anterior corresponde al punto de entrada del intercambio de datos del ejemplo de Recetas Efectivas. Desde Android Studio navegue hasta la clase **MainActivity** en el módulo **wear**. El código presentado a continuación corresponde a la respuesta de la aplicación al comando de voz.

```
MainActivity.java x NotificationUpdateService.java x ProcessProduct.java x
101
102 @Override
103 protected void onActivityResult(int requestCode, int resultCode, Intent data) {
104     if (requestCode == SPEECH_REQUEST_CODE && resultCode == RESULT_OK) {
105         List<String> results = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
106         String spokenText = results.get(0);
107         Log.d("DEBUG", "Google speech result: " + spokenText);
108         if(spokenText.equalsIgnoreCase(Constants.ACTION_BUY_NOW))
109             {
110                 try {
111                     JSONObject json = new JSONObject();
112                     json.put("action", Constants.ACTION_BUY_NOW);
113                     Log.e("ATENCION", "Enviando-> ");
114                     new SendMessage().execute(json.toString());
115                 } catch (JSONException e) {
116                     e.printStackTrace();
117                     Toast.makeText(getApplicationContext(), "Something went wrong",
118                         Toast.LENGTH_LONG).show();
119                 }
120             }
121         } else {
122             Toast.makeText(getApplicationContext(), "Dijiste: " + spokenText +
123                 ". Prueba \"comprar ahora\".",
124                 Toast.LENGTH_LONG).show();
125         }
126     }
127     super.onActivityResult(requestCode, resultCode, data);
128 }
129 }
```

11. Explique el funcionamiento de dicho método. En particular explique la construcción del objeto JSON, cómo se codifican los comandos y como se hace el envío del mismo.
12. ¿Cuál es el método en el módulo **mobile** que recibe dicho JSON? ¿Cómo hace *parsing* del mismo?
13. Utilizando la pregunta anterior como base, escriba brevemente con sus palabras qué decisiones toma el módulo **mobile** cuando llega un *action* distinto a **Constants.ACTION_BUY_NOW**. Mencione dichos otros comandos.
14. Explique cuál es el propósito del *BroadcastReceiver* declarado en la clase *MainActivity* del módulo **wear** en el intercambio de mensajes.

Adición de nuevos requerimientos

15. Modifique el código del módulo **wear** de manera que cuando sea enviada la lista de ingredientes al reloj, la interacción con cada uno de ellos (marcar como “Comprado” o “Pendiente de Compra”) funcione utilizando gestos Swipe (deslizamiento) de la siguiente manera:
 - Para indicar que un ingrediente ha sido comprado se debe deslizar hacia arriba.
 - Para indicar que un ingrediente ha sido marcado como pendiente de compra se debe deslizar hacia abajo.Recuerde que la funcionalidad vía botones debe permanecer intacta.

ENLACES ÚTILES

- 🔗 <http://developer.android.com/training/building-wearables.html>
- 🔗 <http://developer.android.com/training/wearables/data-layer/index.html>
- 🔗 <http://developer.android.com/training/wearables/data-layer/events.html>