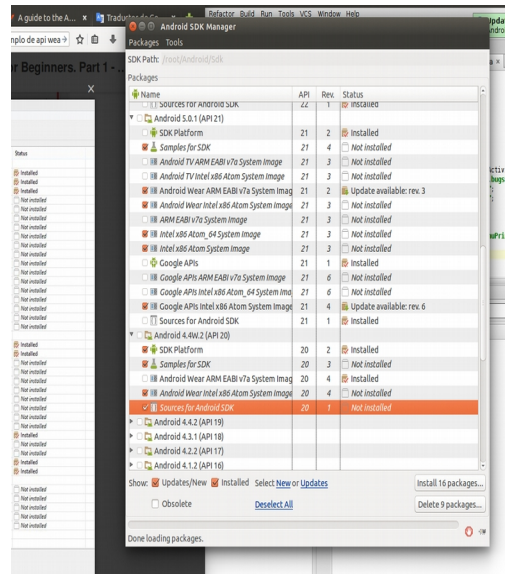
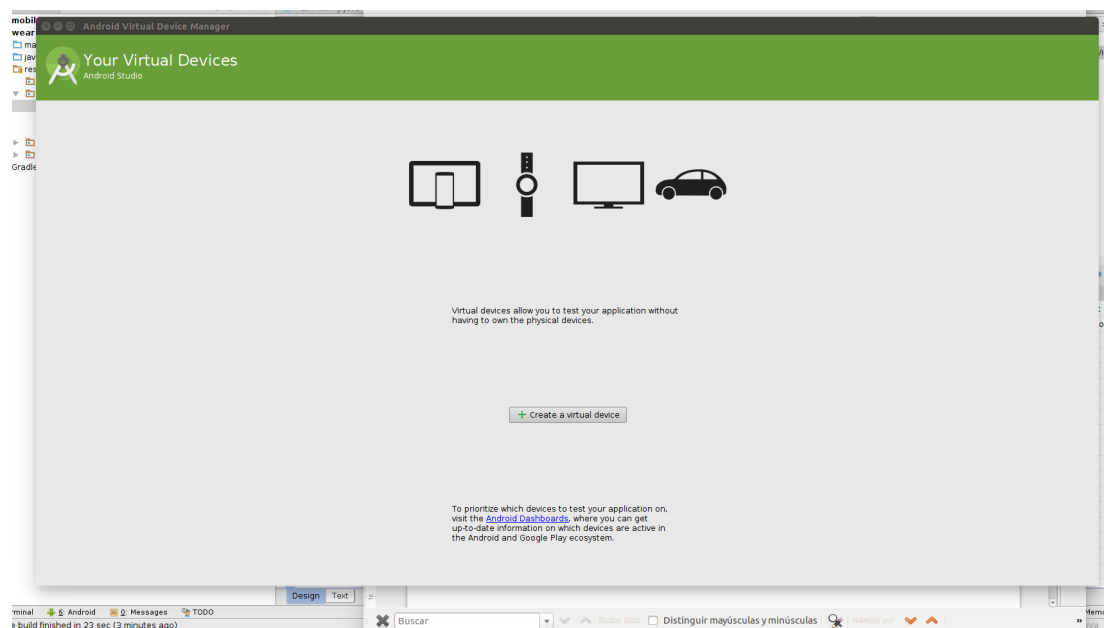


Notificaciones en Android Wear

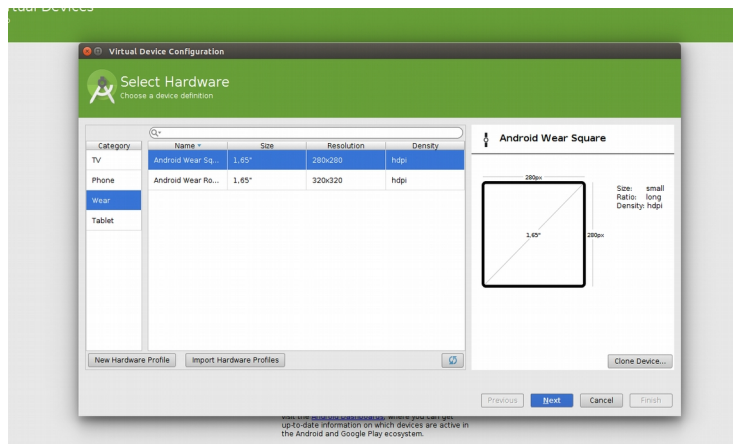
Para poder crear una app que haga uso de la **API Android Wear**, lo primero que necesitamos es instalar en nuestro **Android Studio** las bibliotecas necesarias. Para ello abrimos el **“SDK Manager”** y seleccionamos los siguientes paquetes:



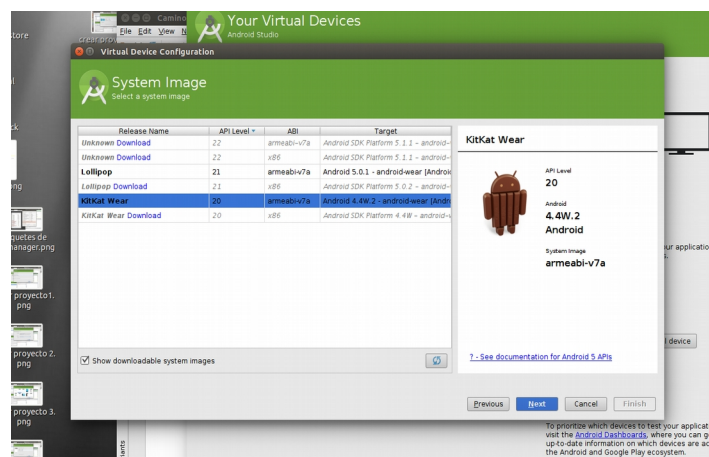
Dependiendo de lo que ya tengáis instalado este proceso puede tardar mas o menos. Si no se dispone de un smartwatch para poder probar la app debemos crear un dispositivo virtual, para esto hacemos click en **“Tools → Android → AVD Manager”**.



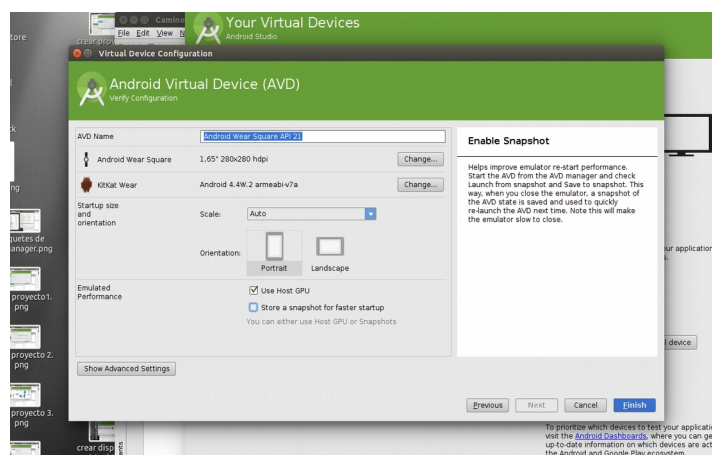
En el interfaz que nos aparece hacemos click en **“Create virtual device”** para añadir un nuevo dispositivo virtual, en el caso que nos ocupa un nuevo *smartwatch*.



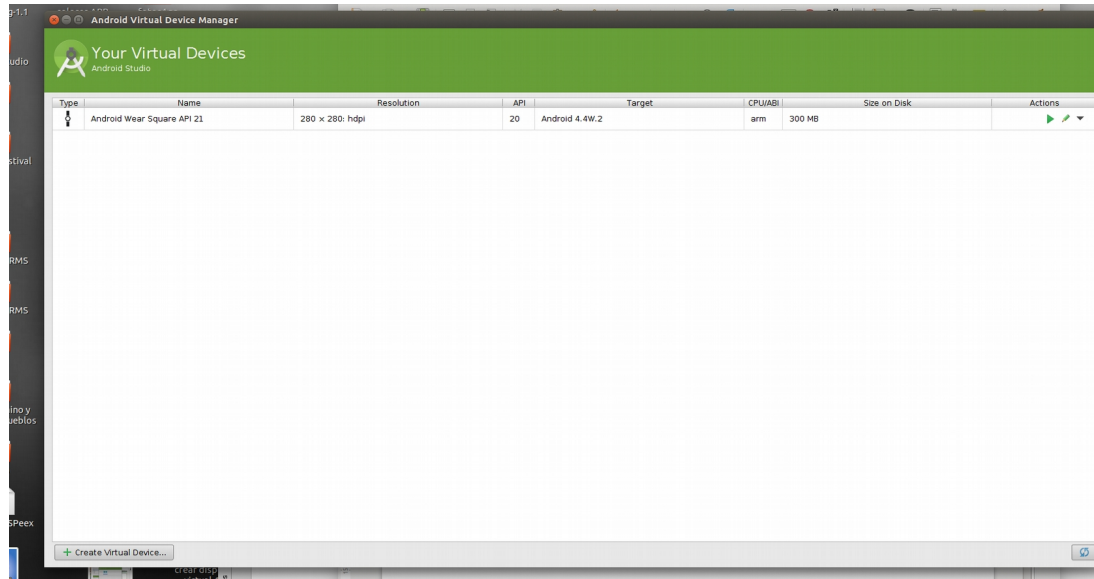
En esta nueva pantalla en la lista de la parte izquierda seleccionamos **“Wear”** y elegimos el tipo de smartwatch que deseamos crear, en nuestro caso el de pantalla rectangular. Una vez seleccionado hacemos click en el botón **“Next”**.



Ahora es el momento de seleccionar la versión de Android que queremos tener en el dispositivo. En este caso elegimos **“KitKat Wear”** cuya versión de API es la 20. Hacemos click en **“Next”**.



Llegados a este punto podemos seleccionar diferentes opciones para nuestra máquina virtual como pueden ser la orientación, la escala, el nombre que tendrá el dispositivo, etc. En este caso dejamos las opciones por defecto y hacemos click en **“Finish”**.



Ahora ya disponemos de una nueva máquina virtual con nuestro dispositivo en la ventana de selección de dispositivos.

En este momento podemos hacer nuestro ejemplo sencillo de app wear, para no alargar mucho este tutorial únicamente voy a insertar el código que genera las notificaciones así como las librerías necesarias para su funcionamiento tanto en *“smartphones”* como en *“smartwatches”*, con la posibilidad de poder interactuar con las mismas mediante botones.

Lo primero que necesitamos para el funcionamiento de las notificaciones en dispositivos *wearable* es insertar en **“Gradle”** la versión de la API con la que deseamos compilar nuestra app, en este caso la mínima versión es: **compile “com.android.support:support-v4:20.0.+”**.

Por otra parte para poder enviar las notificaciones necesitamos importar las siguientes bibliotecas a nuestro código java:

- **android.support.v4.app.NotificationCompat**
- **android.support.v4.app.NotificationManagerCompat**
- **android.support.v4.app.NotificationCompat.WearableExtender**

Por último el código necesario para crear las notificaciones y enviarlas tanto al “*smartphone*” como al “*smartwatch*”, para realizar esto introduce el siguiente código:

```
1 Intent intent = new Intent(Intent.ACTION_VIEW);
2 PendingIntent pendingIntent =
3     PendingIntent.getActivity(this, 0, intent, 0);

4 NotificationCompat.Builder notificacionBuilder =
5     new NotificationCompat.Builder(this)
6     .setSmallIcon(R.drawable.ic_event)
7     .setContentTitle(eventTitle)
8     .setContentText(eventLocation)
9     .setContentIntent(viewPendingIntent)
10    .addAction(R.drawable.ic_map, getString(R.string.map), pendingIntent);
```

Vamos a comentar la función que realiza cada línea:

1. En esta línea se crea un nuevo “**Intent**” que se le pasará al “**PendingIntent**”.
2. Aquí se crea el “**PendingIntent**” que será el que es un tipo de “**Intent**” que espera hasta que se produzca un determinado evento para ejecutarse.
3. Se asigna el “**Intent**” al “**PendingIntent**”.
4. Creamos el constructor de notificaciones.
5. Lo creamos como nuevo constructor del contexto actual.
6. Le asignamos un icono que será el que se muestre tanto en la notificación del *smartphone* como en la del *smartwatch*.
7. Le añadimos título.
8. Le añadimos el texto a mostrar en la notificación.
9. Le asignamos un “**PendingIntent**” a la notificación.
10. Y por último se asigna la acción que se llevará a cabo cuando sea pulsado el botón que se incluye en la misma notificación.