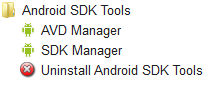
**MANUAL DEL PROGRAMADOR**

1. **HERRAMIENTAS PARA EL DESARROLLO DE LA APLICACIÓN** 
   1. Lo que vamos a hacer es dirigirnos hacia la página de Android developer, cuando estés dentro de ella haz click en la opción SDK, vas a encontrar un cuadro como este:



Podemos descargar una versión para Mac OS X y para Linux, en la sección se Windows hay dos: uno en formato .zip y otra en .exe. Para que se nos haga más fácil seleccionemos el que nos recomienda la misma página (exe) además así no tendremos que descomprimirlo. Si tienes problemas con la instalación puedes leer esta buena guía: (está en inglés, qué bueno que contamos con google translate).

Una vez descargado e instalado debe quedarnos lo siguiente:



* 1. **AVD Manager:** aquí se encuentran las máquinas virtuales que nos permitirán ver la aplicación en funcionamiento, más adelante estaremos utilizando esta herramienta
  2. **SDK Manager:** aquí se encuentran todas las herramientas ya instaladas y las que no hemos instalado, cada carpeta es una versión de Android. Dentro de ellas se encuentra el SDK que es básicamente el elemento que necesitamos para programar.
  3. **Plugin Adt Para Eclipse**

Un Plugin es un complemento que añade función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal.

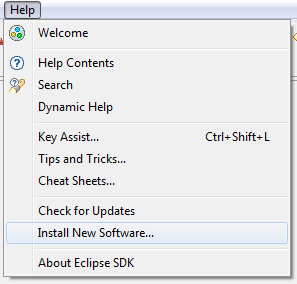
Sin salirnos de la página “android developer” vamos a seleccionar el ADT 18.0.0 (es un plugin para el IDE de Eclipse que está diseñado para darle un ambiente potente e integrado) en el menú izquierdo de la página:

http://l4c.me/uploads/sin-titulo-78-1336877591_full550.png

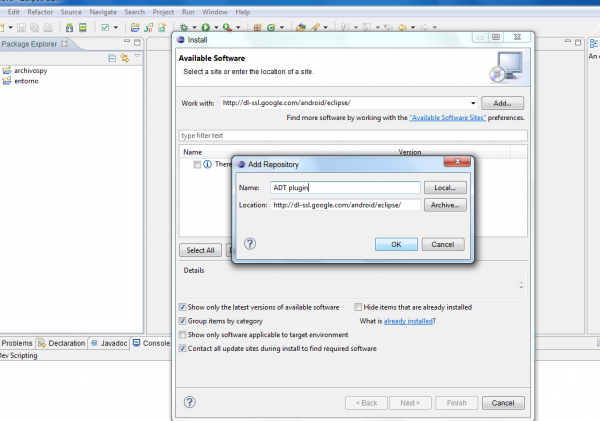
Luego se nos mostrarán las instrucciones para añadirlo en eclipse, en una parte de la página vamos a encontrar la siguiente dirección:

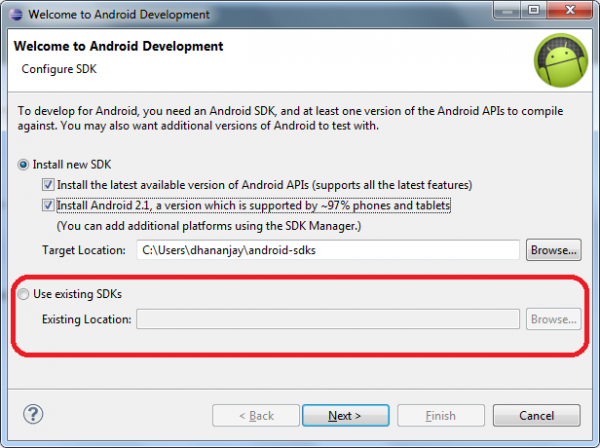
**http://dl-ssl.google.com/android/eclipse/**

Copiemos la dirección y ejecutemos eclipse y dirijámonos a help > install new software



Y pegamos la dirección en work with, luego seleccionamos “add”, se nos abre una ventana pidiéndonos el nombre que le pondremos al plugin en mi caso yo escribí simplemente “ADT plugin”:



Hacemos click en ok y Eclipse estará verificando y llamando a la dirección que le hemos indicado. Todos los elementos que aparecerán luego deben estar seleccionados (select all); damos click en next > next > aceptamos las condiciones de uso > finish. Eclipse se va a reiniciar y nos aparecerá una ventana como esta:

Nos vamos a la opción use existing SDKs y ubicamos la ruta donde se instaló, de lo contrario usamos install new SDK y marcamos la segunda casilla. Terminamos el proceso y listo!!

1. **Crear un dispositivo virtual**

Es algo parecido a usar virtualBox. En eclipse ahora van a aparecer dos iconos en el menú:

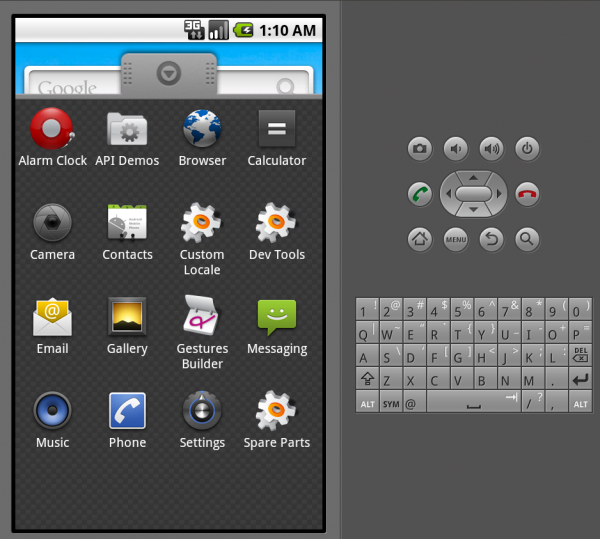
El primero es el AVD manager y el segundo es SDK manager, escojamos el segundo para crear una máquina virtual > click en new. Luego nos aparecerán las características del dispositivo:

**Name:** escribimos el nombre

**Target:** elegimos la versión que vamos a usar ej: 2.1

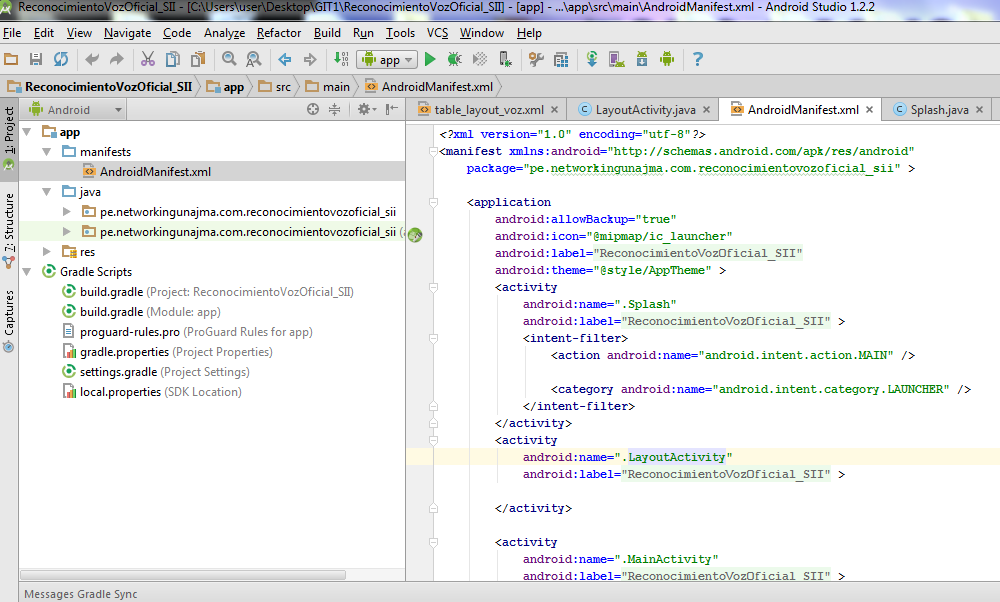
**SD card:** elegimos la cantidad de espacio que ocupa, podemos poner 1GIB

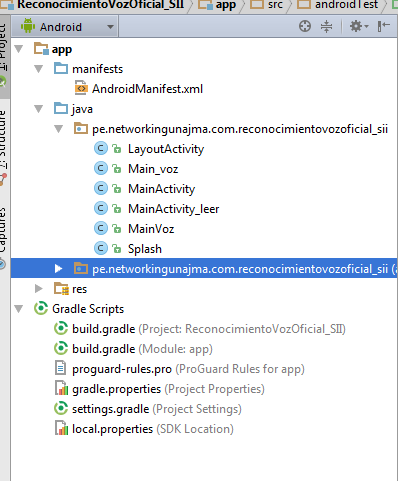
**Hardware:** si damos click en new podemos añadirle soporte para funcionamientos como: acelerometrer, soporte para touch-screen, e.t.c

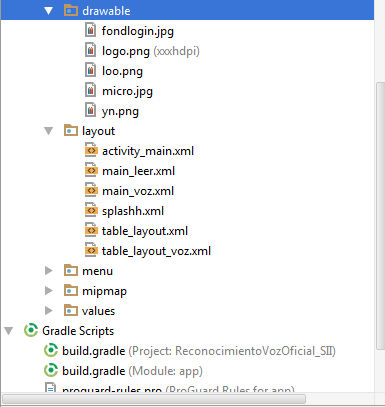
Para finalizar click en create AVD, si queremos ver en funcionamiento la máquina virtual, click en start:

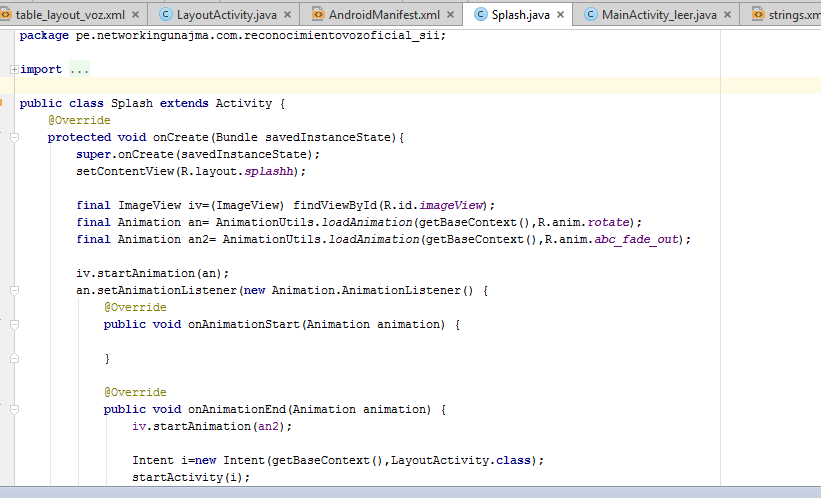
Ahora, ya podemos empezar a programar y crear un nuevo proyecto android.

**“Desarrollo la aplicación Reconocimiento\_de\_Voz\_Oficial”**

**3.1 Descripción del proyecto:**

**3.2**

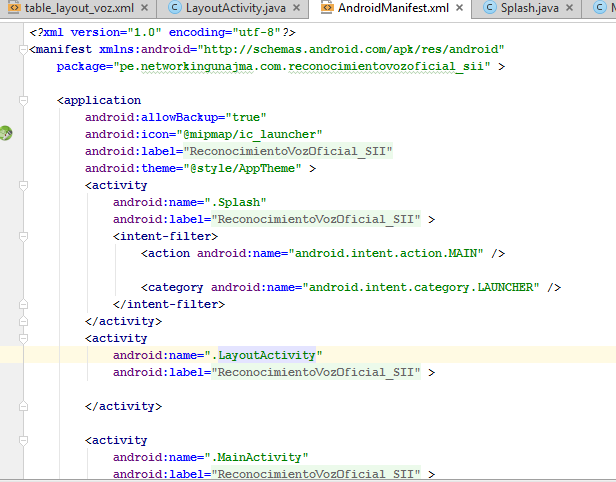
3.3

3.4 Splash.java es el encargado de gestionar la pantalla de bienvenida de la aplicación.

3.5 AndoridManifest.xml

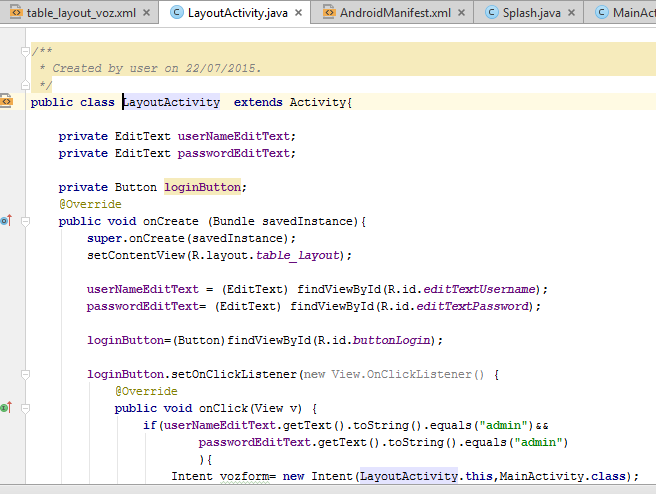
Todas las aplicaciones deben contener este archivo por convención. El nombre debe permanecer intacto, ya que se usa como referencia para el parsing de nuestra aplicación. El nodo raíz de este documento se representa con la etiqueta <manifest> y por obligación debe contener un hijo de tipo <application>.

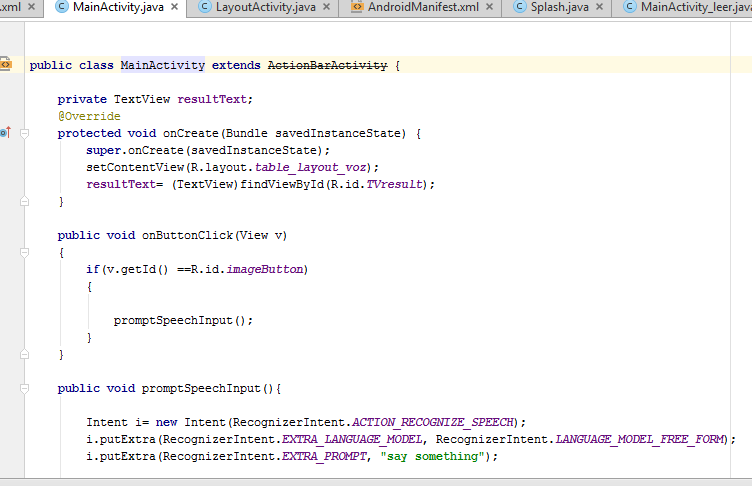
La etiqueta <application> representa como estará construida nuestra aplicación. Dentro de ella definiremos nodos referentes a las actividades que contiene, las librerías incluidas, los Intents, Providers, y demás componentes.



3.6 LayoutActivity.java

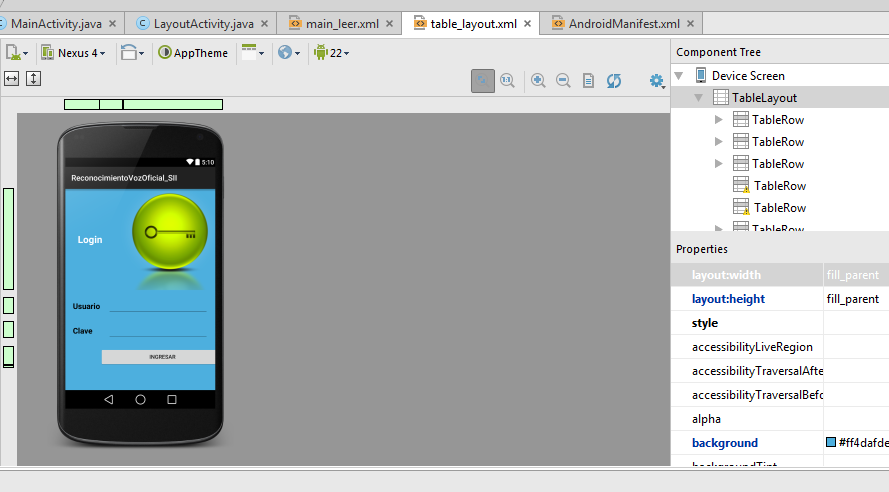
Contiene el desarrollo de la parte del login de la aplicación .

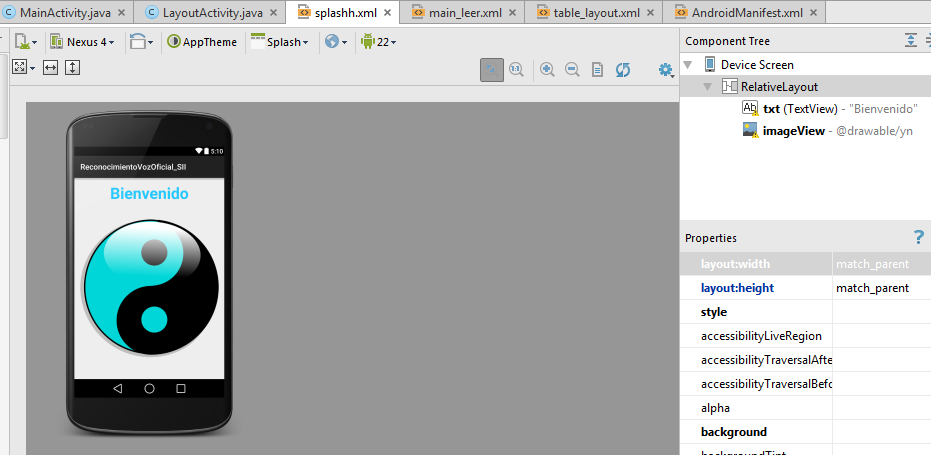


3.7 MainActivity.java contiene el desarrollo de la aplicación principal

**4.0 Los Layout**

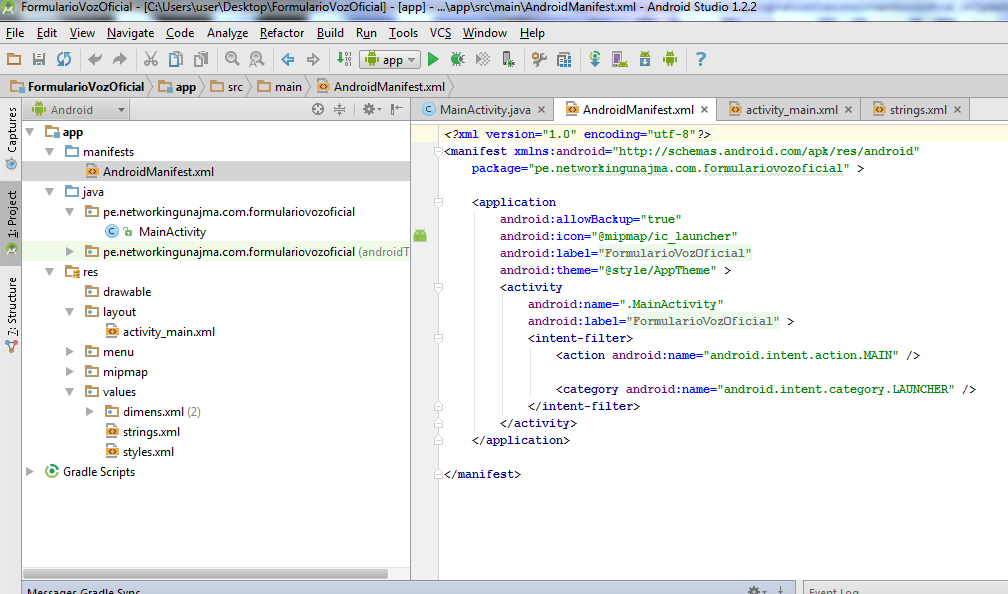
**4.1 Table\_layout contiene la interfaz del desarrollo del login de la aplicación**

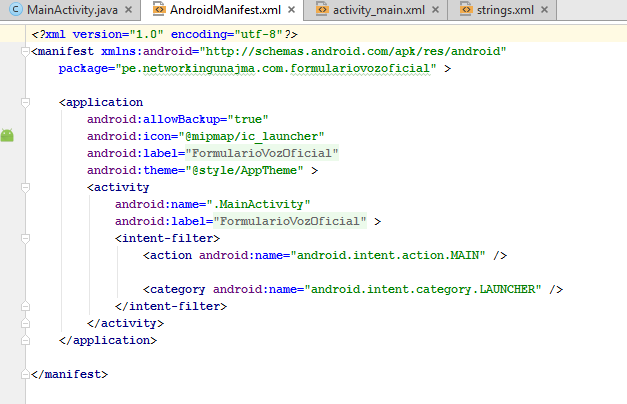
****

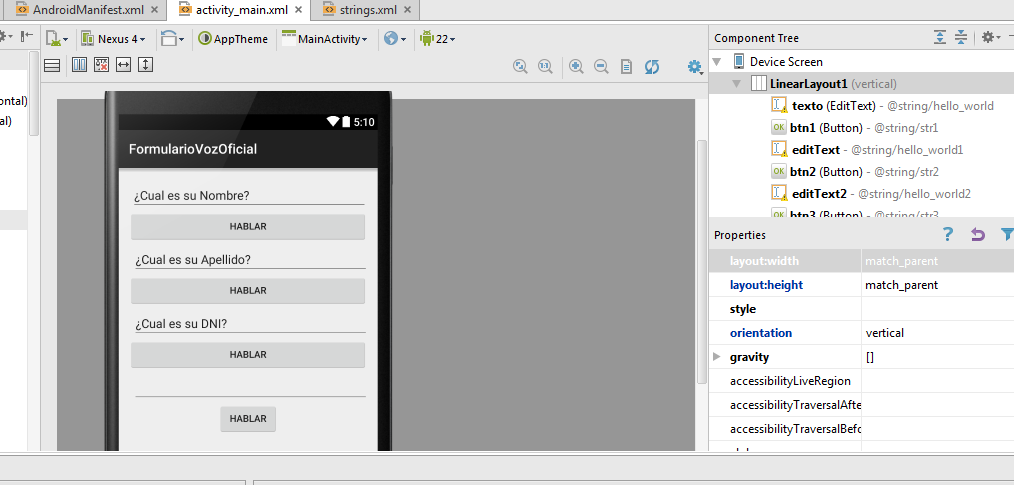
4.2 splashh.xml contiene la interfaz de desarrollo de bienvenida.

5.0 **DESARROLLO PARTE LEER TEXTO:**

5.1 La estructura de la aplicación

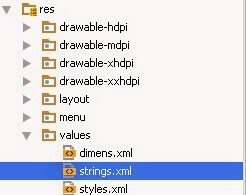


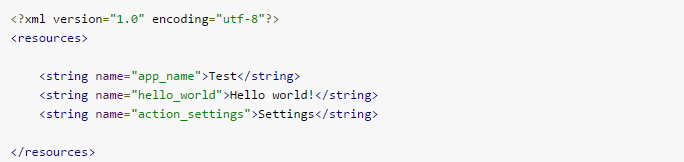
5.2 El AndroidManifest 

5.3 Activity\_main .xml 

6.0 **STRINGS.XML**

Dentro de la carpeta”res” encontraremos todos aquellos recursos tercerizados para nuestra aplicación. Esta práctica de excluir los atributos de la aplicación a través de archivos externos, permite reducir la complejidad de diseño en las interfaces.



Uno de los recursos mas relevantes es el archivo strings.xml que se encuentra dentro de la subcarpeta values. Este fichero almacena todas las cadenas que se muestran en los widgets(controles, formas, botones, vistas, etc) de nuestras actividades.

Si abres el archivo, verás que se encuentran tres nodos del tipo <string>: “app\_name”, “action\_settings” y “hello\_world”.