**Sesión 1-02 Clase del 2 de octubre**

En esta sesión vamos a realizar unas operaciones de preparación del IDE y de sus herramientas.

1.- Iniciamos y utilizamos la herramienta SDK Manager:

* Instalamos otros dos SDK (**Android Q y Android Marshmallow**). Si es necesario, actualizamos el SDK que tenemos instalado actualmente.
* Comprobamos cual es la versión del emulador instalada. Si es necesario, actualizamos la versión.
* Si no está instalado, instalamos **Google Play services** para que se puedan utilizar servicios de Google Play en nuestras aplicaciones (por ejemplo Google maps).
* Si no está instalado, instalamos **Google USB drive** para que se puedan ejecutar nuestras aplicaciones en dispositivos reales conectados por USB al ordenador.

2.- Creamos y configuramos un AVD. Debe usar el SO Android Q.

3.- Iniciamos con AVD Manager el dispositivo emulado. Compilamos y ejecutamos la aplicación del proyecto sobre el dispositivo emulado.

4.- Vemos como se puede probar en un dispositivo real conectado por USB una app desarrollada en Android Studio.

5.- Desactivamos actualizaciones automáticas del IDE y de sus componentes**.**

Para desactivar las actualizaciones del IDE y pluggins, **File > Settings > Appearance & Behavior > System Settings > Updates**

6.- Retomamos el estudio de código para desarrollar el diseño de la pantalla en **MainActivity.kt.** Agregamos una función **Composable** para diseñar un botón (Button). Dentro del Button vamos a poner un texto SALUDAR.

7.- Agregamos en Main la llamada a la función desarrollada en 6. Vemos lo que ocurre.

8.- Establecemos que la funcionalidad del Button sea escribir en terminal un Log con el texto “Bienvenido a Android!” y probamos.

9.- Vamos a ver algo sobre los propiedades y modificadores de componentes). Podemos trabajar sobre el mismo proyecto o, si queremos, iniciar uno nuevo.

10.- En **setContent** de **onCreate,** hacemos la llamada a una nueva función **Content().** Si es el caso, eliminamos todo el código sobrante.

La función **Content()** pintará contenido de la pantalla (es composable). Llamará a una función que pinta un texto (Text). Probamos varias propiedades de Text:

**fontSize,  
color,  
fontWeight,  
softWrap,  
maxLines,  
textAlign,  
overflow,  
fontFamily,  
lineHeight,  
letterSpacing,  
onTextLayout**

11.- Ahora desde Content llamamos a otra función que pinta el texto recibido como parámetro:

El componente Text debe usar un tamaño de fuente de 20 sp, el color de la letra será verde y el texto debe estar alineado a la derecha en el componente.

Le aplicamos modificadores al componente Text:

fillMaxWidth(),

background(),

padding(),

clickable