



cpi'fp

Los Enlaces

PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES

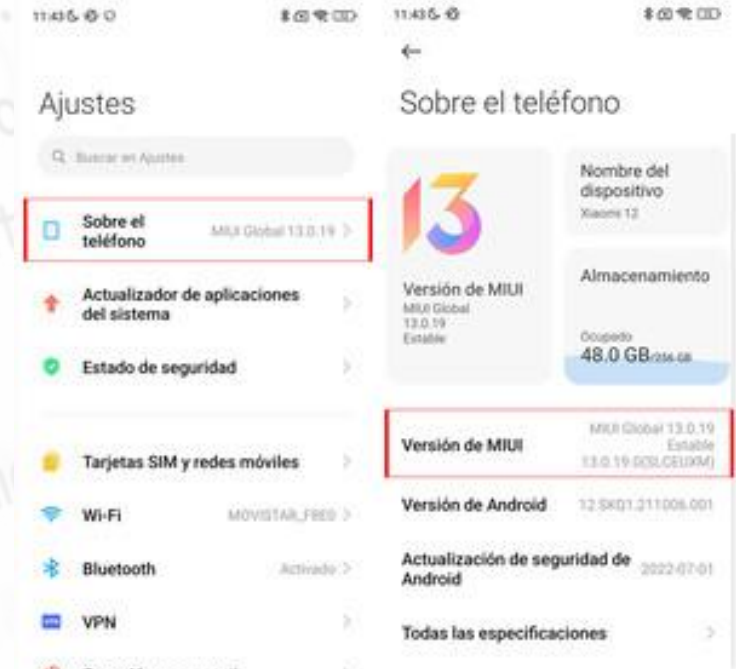
2º DESARROLLO DE APLICACIONES MULTIPLATAFORMA

TEMA 4. CONEXIÓN A DISPOSITIVO FÍSICO

T4. CONEXIÓN A DISPOSITIVO FÍSICO

0. MODO DESARROLLADOR

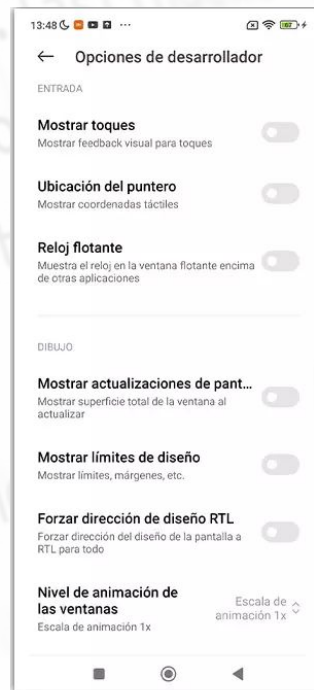
Dependiendo del modelo de teléfono que tengamos, activar las opciones de desarrollador se hará de una manera u otra. Lo normal es acceder dentro de *Ajustes* al apartado *Sobre el teléfono* o *Acerca del dispositivo* y pulsar varias veces en *Versión...* o *Info de software* o similar. También puede ser a través de *Privacidad y seguridad* activando la opción *Modo desarrollador*.



0. MODO DESARROLLADOR

Una vez activado, encontraremos un nuevo menú *Opciones de desarrollador* o simplemente *Desarrollador* que puede estar dentro de *Ajustes adicionales* o en el apartado *Seguridad*.

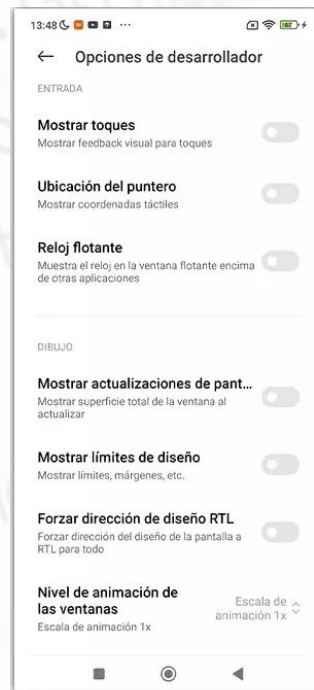
Si activamos la depuración USB estaremos permitiendo el acceso al dispositivo cuando se esté conectado a un ordenador con un cable USB.



0. MODO DESARROLLADOR

También existe, desde la versión de Android Studio Bumblebee, conectarlo a un dispositivo físico a través de WiFi mediante la opción de desarrollador *Depuración inalámbrica*.

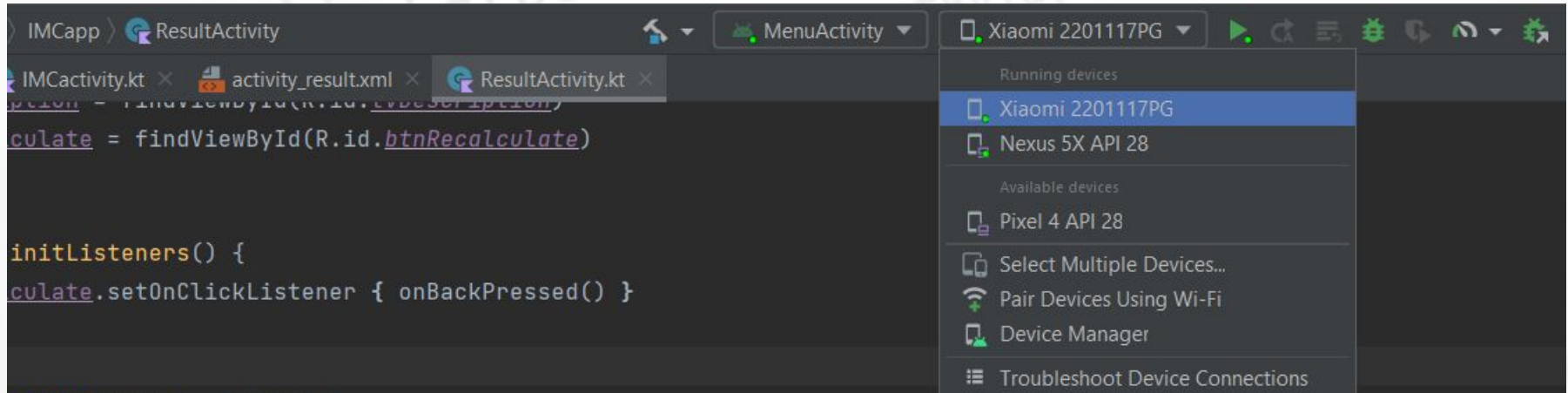
Lo recomendable es conectarse a dispositivos de confianza a través de USB o deshabilitar la depuración inalámbrica cuando no estemos conectados a una red segura.



T4. CONEXIÓN A DISPOSITIVO FÍSICO

1. CONEXIÓN DESDE ANDROID STUDIO

Actualmente Android Studio detectará de forma automática nuestro dispositivo una vez hayamos indicado en el mismo que el ordenador al que nos conectamos es de confianza.



T4. CONEXIÓN A DISPOSITIVO FÍSICO

1. CONEXIÓN DESDE ANDROID STUDIO

Si el teléfono no es detectado, es posible que sea necesario instalar un controlador USB de OEM (Original Equipment Manufacturer). En la documentación oficial de Android Studio podemos encontrar una tabla con algunos enlaces a sitios oficiales donde descargar los drivers:

<https://developer.android.com/studio/run/oem-usb?hl=es-419#Drivers>

También se pueden encontrar en Google rápidamente. Por ejemplo, para Xiaomi:

<https://xiaomidriver.com/>

2. PROBLEMA CON DecimalFormat()

Es posible que la aplicación IMCapp dé un error
`java.lang.NumberFormatException: For input string: "48,61"` ,
el cual se puede solucionar con el siguiente código:

```
private fun calculateIMC():Double {  
    val dfs = DecimalFormatSymbols()  
    dfs.decimalSeparator = '.'  
    val df = DecimalFormat("#.##")  
    df.decimalFormatSymbols = dfs  
    val imc = currentWeight / pow(currentHeight.toDouble()/100,  
2.0)  
    return df.format(imc).toDouble()  
}
```



3. ICONO DE APLICACIÓN

En el siguiente enlace se muestra la manera de cambiar el icono de una aplicación:

<https://developer.android.com/codelabs/basic-android-kotlin-compose-training-change-app-icon?hl=es-419#0>

Lee toda la documentación y realiza el codelab. Puedes elegir cualquier otra imagen distinta a la suministrada en el codelab pero ten en cuenta las indicaciones sugeridas para añadir un icono de aplicación.

