

Tutoría 2 UX/UI Grupo X

Programación para Dispositivos Móviles

Componentes de una aplicación Android, Interfaz gráfica, Experiencia de usuario

Agenda

- Objetivos de la tutoría
- Componentes de una aplicación Android. Conceptos esenciales.
- Emparejamiento de dispositivo físico Android con Android Studio.
- Diseño de ejercicio en Figma y prototipo (Conversión de temperaturas).
- Despedida

Fundamentos de una aplicación móvil

Para escribir aplicaciones de Android, es posible usar los lenguajes Kotlin, Java y C++. Las herramientas de Android SDK compilan tu código, junto con los archivos de recursos y datos, en un APK: ***un paquete de Android, que es un archivo de almacenamiento con el sufijo .apk.***

Un archivo APK incluye todos los contenidos de una aplicación de Android y es el archivo que usan los dispositivos con tecnología Android para instalar la aplicación.

Componentes de la aplicación - Actividades

Una actividad es el punto de entrada de interacción con el usuario. Representa una pantalla individual con una interfaz de usuario.

Por ejemplo, ***una aplicación de correo electrónico*** tiene una actividad que muestra una lista de los correos electrónicos nuevos, otra actividad para redactar el correo electrónico y otra actividad para leerlo.

Si bien las actividades funcionan juntas para ofrecer una experiencia de usuario uniforme en la aplicación de correo electrónico, cada una es independiente de las demás.

De esta manera, una aplicación diferente puede iniciar cualquiera de estas actividades (si la aplicación de correo electrónico lo permite).

Nota: Las actividades se implementan como subclases de la clase Activity

Componentes de la aplicación - Servicios

Un servicio es un punto de entrada general que permite mantener la ejecución de una aplicación en segundo plano por diversos motivos. Es un componente que se ejecuta en segundo plano para realizar operaciones de ejecución prolongada o para realizar tareas de procesos remotos. Un servicio no proporciona una interfaz de usuario.

Por ejemplo, un servicio podría reproducir música en segundo plano mientras el usuario se encuentra en otra aplicación, o podría capturar datos en la red sin bloquear la interacción del usuario con una actividad. Otro componente, como una actividad, puede iniciar el servicio y permitir que se ejecute o enlazarse a él para interactuar.

Nota: Un servicio se implementa como una subclase de Service.

Componentes de la aplicación - Servicios

Servicios enlazados

Los servicios enlazados se ejecutan porque otra aplicación (o el sistema) indicó que quiere usarlos. Básicamente, lo que sucede es que un servicio le brinda una API a otro proceso.

Por lo tanto, el sistema sabe que hay una dependencia entre estos procesos. En consecuencia, si el proceso A está enlazado a un servicio en el proceso B, sabe que debe mantener funcionando el proceso B (y el servicio correspondiente) para el proceso A.

Además, si el proceso A es de interés para el usuario, también sabe que debe tratar el proceso B teniendo esto en cuenta.

Componentes de la aplicación – Receptores de emisiones

Un receptor de emisión es un componente que posibilita que el sistema entregue eventos a la aplicación fuera de un flujo de usuarios habitual, lo que permite que la aplicación responda a los anuncios de emisión de todo el sistema.

Dado que los receptores de emisión son otro punto bien definido de entrada a la aplicación, el sistema puede entregar emisiones incluso a las aplicaciones que no estén en ejecución.

Por ejemplo, una aplicación puede programar una alarma para publicar una notificación sobre un evento futuro destinada al usuario. Al entregar dicha alarma al receptor de emisión de la aplicación, no hace falta que dicha aplicación siga ejecutándose hasta que se active la alarma.

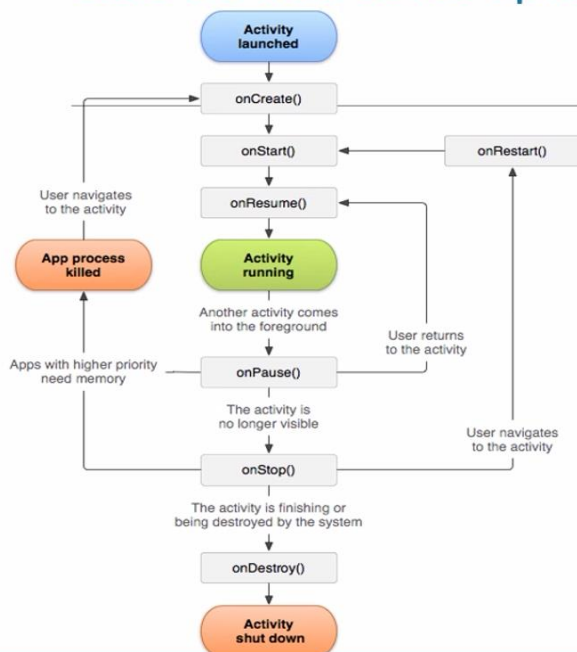
Componentes de la aplicación – Receptores de emisiones

Las aplicaciones también pueden iniciar emisiones, por ejemplo, para avisarles a las demás aplicaciones que se descargaron datos al dispositivo y que están disponibles para el uso.

Si bien los receptores de emisión no exhiben una interfaz de usuario, pueden crear una notificación de la barra de estado para alertar al usuario cuando se produzca un evento de emisión. Sin embargo, por lo general, un receptor de emisión es simplemente una puerta de enlace a otros componentes y está destinado a realizar una cantidad mínima de trabajo. Por ejemplo, podría programar un servicio **JobService** para que realice algunas tareas en función del evento con **JobScheduler**.

Nota: Un receptor de emisión se implementa como una subclase de `BroadcastReceiver` y cada receptor de emisión se entrega como un objeto `Intent`.

Ciclo de vida de las aplicaciones (Android)



Acá veremos un ejemplo en vivo en un proyecto Android que muestra como la aplicación cambia de estados mientras interactuamos con nuestro teléfono.

Ver proyecto de Ejemplo **EstadosAndroid**


```
package com.tutoria2.estadosandroid

> import ...

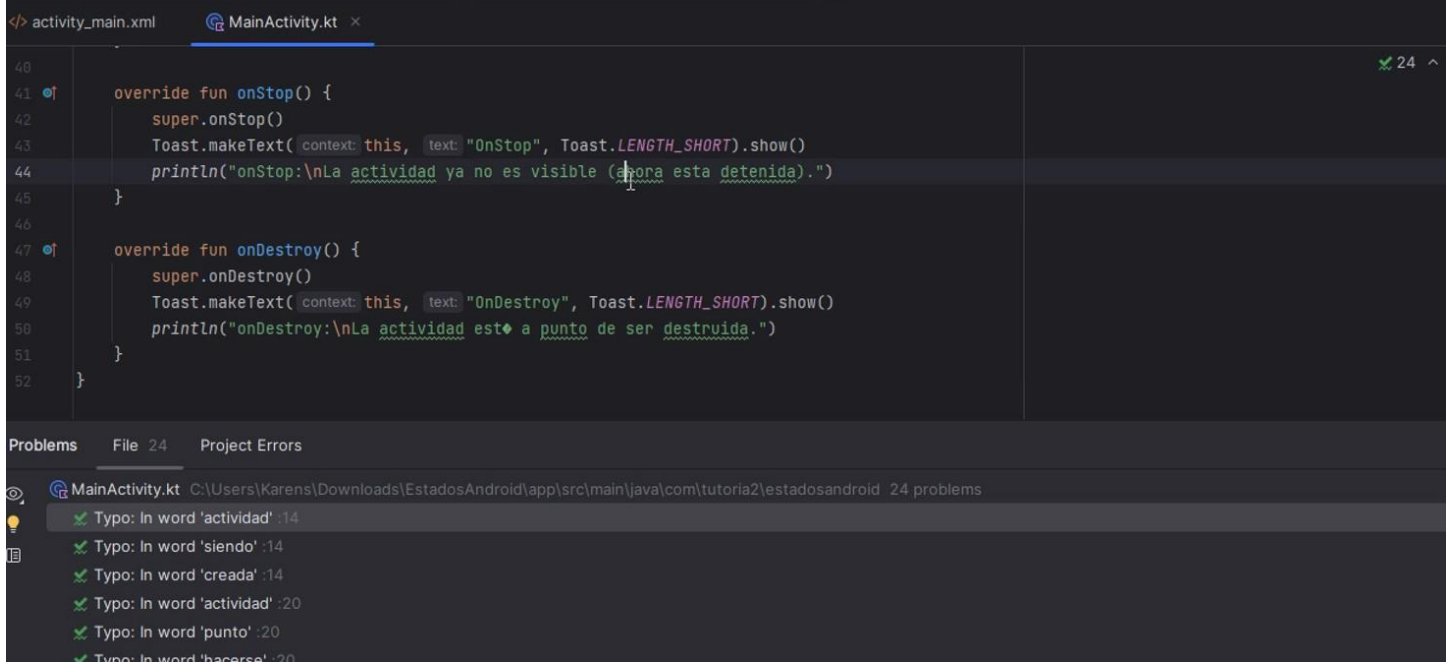
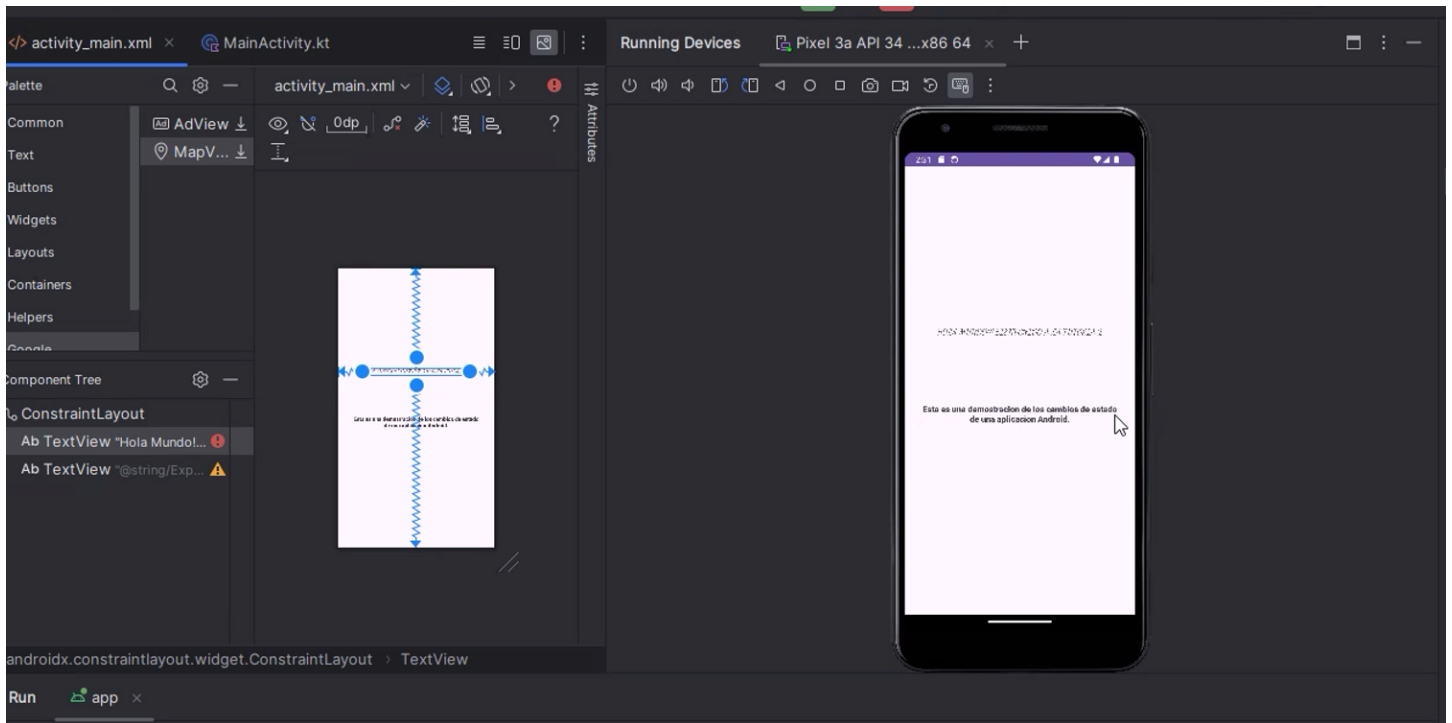
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        {
            super.onCreate(savedInstanceState)
            setContentView(R.layout.activity_main)
            Toast.makeText(context: this, text: "OnCreate", Toast.LENGTH_SHORT).show()
            println("onCreate:\nLa actividad esta siendo creada.")
        }

        override fun onStart() {
            super.onStart()
            Toast.makeText(context: this, text: "OnStart", Toast.LENGTH_SHORT).show()
            println("onStart:\nLa actividad est a punto de hacerse visible.")
        }

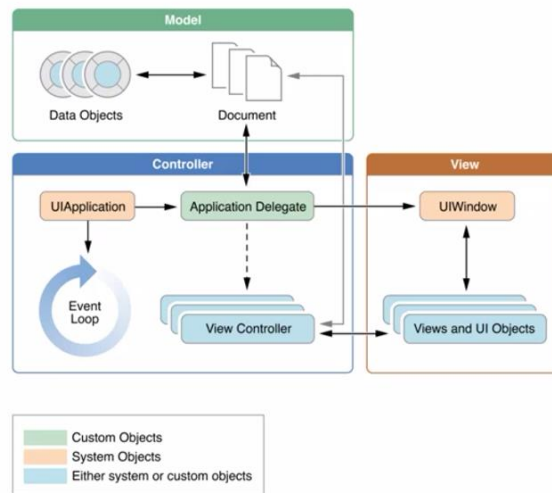
        override fun onResume() {
            super.onResume()
            Toast.makeText(context: this, text: "OnResume", Toast.LENGTH_SHORT).show()
            println("onResume:\nLa actividad se ha resumido ya es visible.")
        }

        override fun onRestart() {
            super.onRestart()
        }
    }
}
```

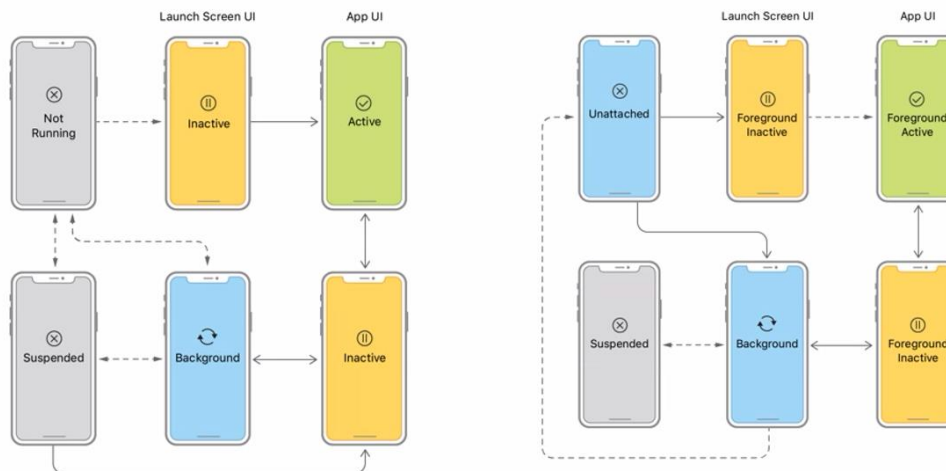
Emulador pixel API 24 tanto el gradle como la virtual



Ciclo de vida de las aplicaciones (iOS)



Ciclo de vida de las aplicaciones (iOS)



El archivo de manifiesto (AndroidManifest.xml)

Para que el sistema Android pueda iniciar un componente de la aplicación, debe reconocer la existencia de ese componente leyendo el archivo de manifiesto de la aplicación (AndroidManifest.xml).

La aplicación debe declarar todos sus componentes en este archivo, que debe encontrarse en la raíz del directorio de proyectos de la aplicación.

El manifiesto puede hacer ciertas cosas además de declarar los componentes de la aplicación, por ejemplo:

- Identificar los permisos de usuario que requiere la aplicación, como acceso a Internet o acceso de lectura para los contactos del usuario.
-