

Arquitectura de Android

La arquitectura de Android es el conjunto de componentes, frameworks y herramientas que permite el desarrollo y ejecución de aplicaciones móviles en el sistema operativo Android. Esta arquitectura se caracteriza por su flexibilidad, modularidad y capacidad de adaptación a diferentes dispositivos.



Presenta: M. en C. Gabriel Hurtado Avilés



Componentes Principales

Los componentes principales de la arquitectura de Android son Activities, Fragments, Services, Broadcast Receivers y Content Providers. Cada uno desempeña un rol específico en la funcionalidad de una aplicación.

1 Activities

Las Activities son las pantallas de la aplicación que interactúan con el usuario. Son las unidades básicas de la interfaz de usuario.

2 Fragments

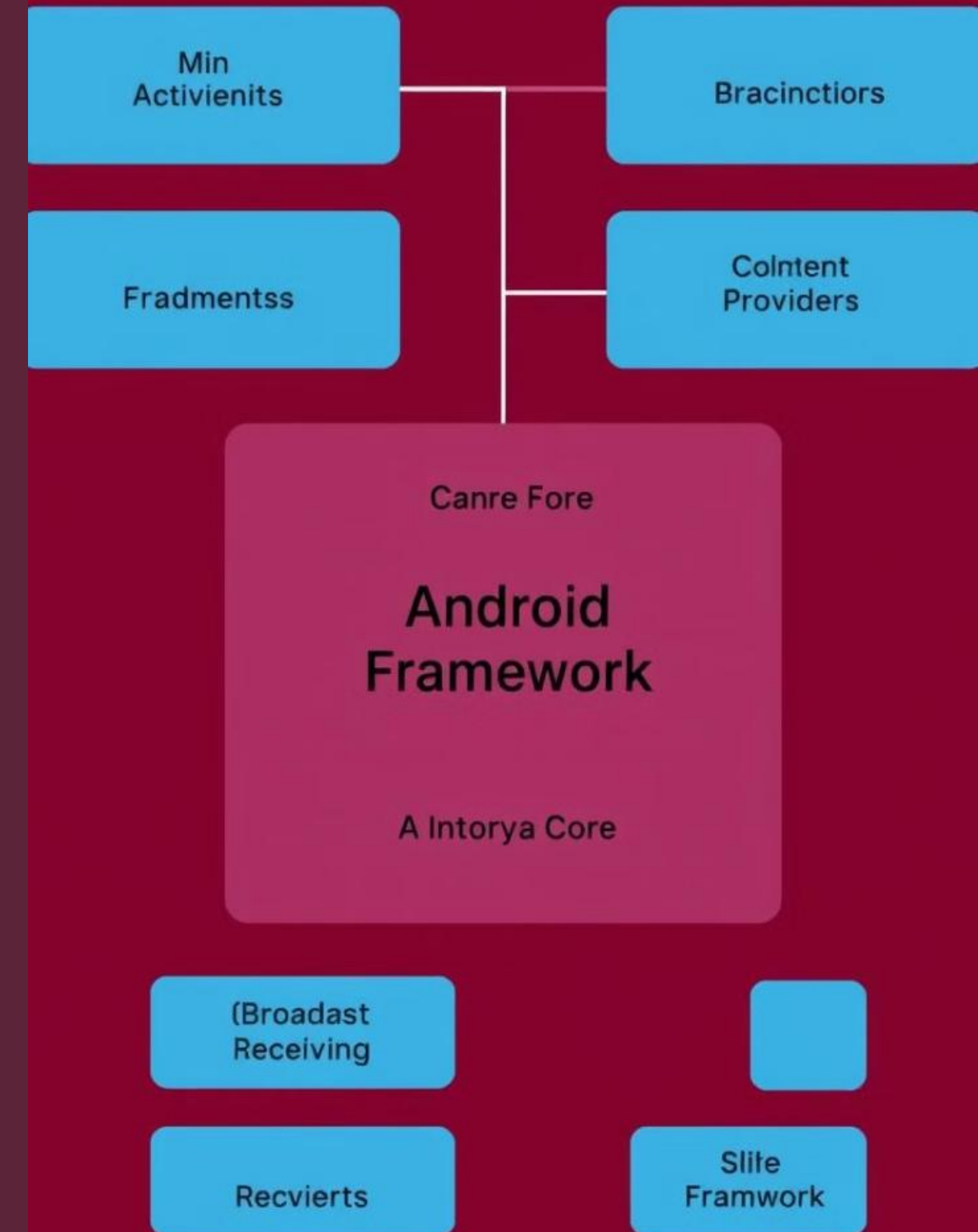
Los Fragments son partes reutilizables de una Activity que permiten crear interfaces más flexibles y modulares.

3 Services

Los Services se ejecutan en segundo plano y realizan tareas que no requieren una interacción directa con el usuario.

4 Broadcast Receivers

Los Broadcast Receivers reciben y procesan eventos del sistema o de otras aplicaciones.



Sistema de Archivos y Estructura de Directorios

El sistema de archivos de Android organiza los datos de la aplicación, el sistema operativo y los archivos multimedia. La estructura de directorios es jerárquica y permite el acceso y la gestión de archivos.

Directorio Raíz

El directorio raíz contiene las carpetas principales del sistema, como "data", "media" y "system".

Directorio de Datos

La carpeta "data" contiene los datos de las aplicaciones, incluyendo bases de datos, preferencias y archivos temporales.

Directorio de Multimedia

La carpeta "media" almacena archivos multimedia como fotos, videos y audios.

Ciclo de Vida de una Aplicación Android

El ciclo de vida de una aplicación Android describe las diferentes etapas por las que pasa la aplicación durante su ejecución. La aplicación puede entrar en diferentes estados, como activo, inactivo o detenido, y el sistema Android gestiona estas transiciones.

onCreate()

Se ejecuta al iniciarse la aplicación. Inicializa la Activity.

1

onStart()

La Activity se hace visible al usuario.

2

onResume()

La Activity está en primer plano y puede interactuar con el usuario.

3

onPause()

La Activity pierde el foco, pero sigue en ejecución.

4

onStop()

La Activity ya no es visible al usuario.

5

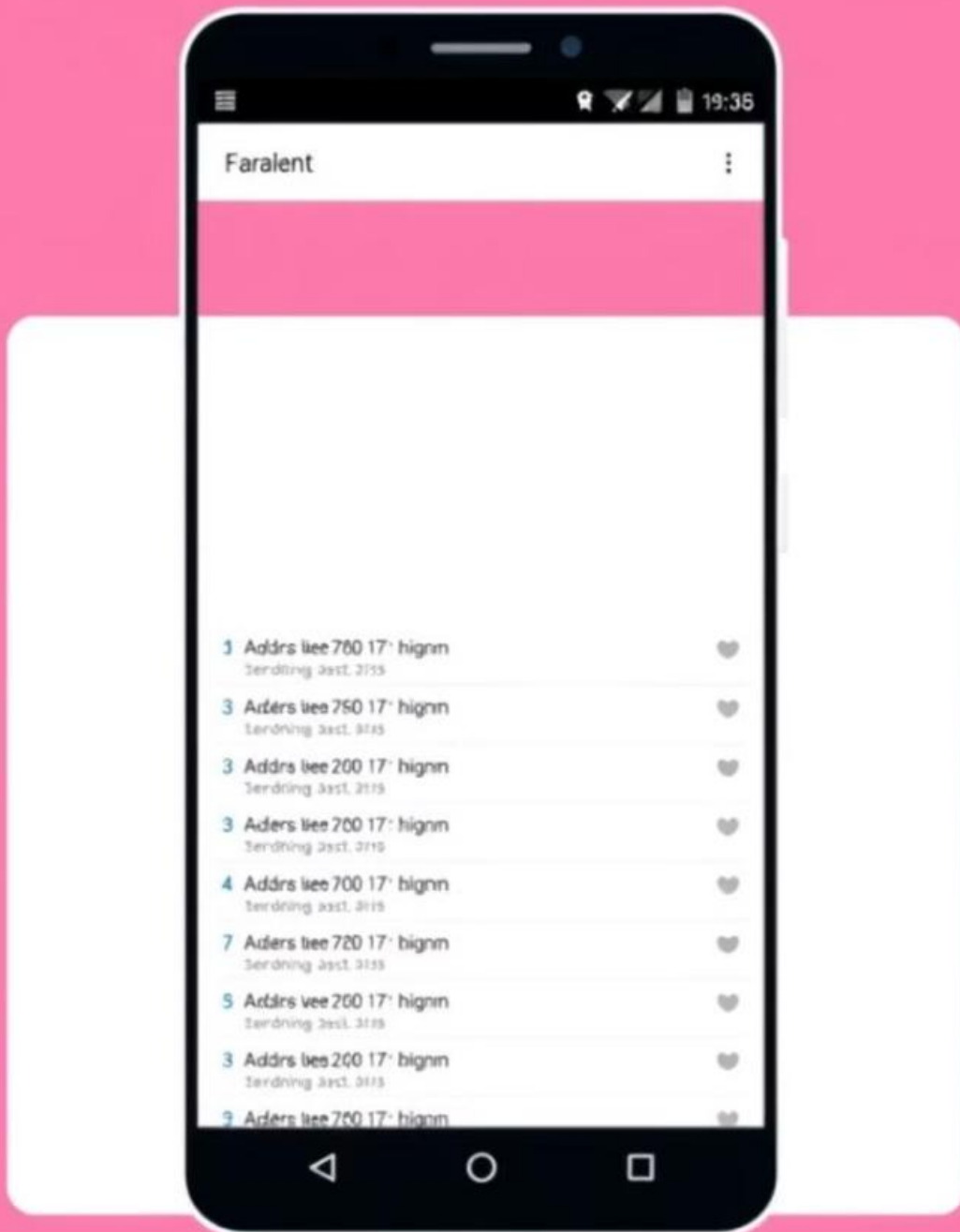
onDestroy()

La Activity es destruida.

6

Activities y Fragments

Las Activities son las unidades básicas de la interfaz de usuario de Android. Los Fragments son partes reutilizables que se pueden integrar en una Activity.



Activities

1

Las Activities son las pantallas principales de la aplicación, como la pantalla de inicio o la pantalla de detalles de un producto.

Fragments

2

Los Fragments son componentes reutilizables que pueden ser incluidos en una Activity.

Integración

3

Un Fragment puede ser incluido en una Activity para agregar una funcionalidad específica.

Gestión de Recursos y Layouts

Los recursos de Android incluyen imágenes, cadenas de texto, estilos y otros archivos que se utilizan para construir la interfaz de usuario. Los Layouts definen la estructura y organización de los elementos de la interfaz.

Recursos	Layouts
Imágenes	LinearLayout
Cadenas de texto	RelativeLayout
Estilos	ConstraintLayout

```
tranL_:- >g {
  anflase: <[]
  cronetalt: lunporal <1>:
    recoent: <iee_nnofle antutorls);
    secuice: <iee-any boons pogrettitled);
    recure: <>:
    secha/ler< 5!>
    tant: log-igs-<=>:
    inta/losy.retvacliae= 6>
    tech/llake ant-6oonetes 6!>
    inta/lasdriye= 6:
    vest.tye < 5>
    ianant: way = butt-ive=_1:
    securdes agradult poobilc(aser(wan.wire vattones:rsecisce):):
    constonons ovice-(9):
      varuse [aesland igecrwpt.[atethengiscstfc_ancriun.f);
      suttve:(othy conner caopu(atec_pottret);
    ..'väll: [arceblotyectserd))
    rantes and oooliog f(tern(pordè);
    reession tblcflang detess(boflace:f);
    rapses pottanl wppoot: yog/actlectivd arscsess brate dotidiard);
  }>
  recorsatd:_gaopstfuseordf); {
    (aast-and -areperd);
    recorscoup:/aftliyctiyneMbl.(tgraist)
    (aasse.apo <lt);
    rcrestation with wwertJUbityaded);
  secuides < 2:
  reconlstavingine((apce(acard:
    rgrestation and litg:(noocd(scross.
    rrests-(andsage bogalloy/wecrior taccidenM.anbne.itcb:jsts;"watlegive
    trest: and hylce {
      "thtinttes long"emptipnethelared.com_nc_echtaectvectisl);
    }):
    crotice cotee= {::
      rse(aging nitle nest videy);
    }>
  }>

  vatags app of imtarrisees Ind kakr );
  seavilve;
}>
```

Comunicación entre Componentes

Los componentes de Android pueden comunicarse entre sí utilizando diferentes mecanismos. Estos mecanismos permiten la coordinación y el intercambio de datos entre las diferentes partes de la aplicación.

Intents

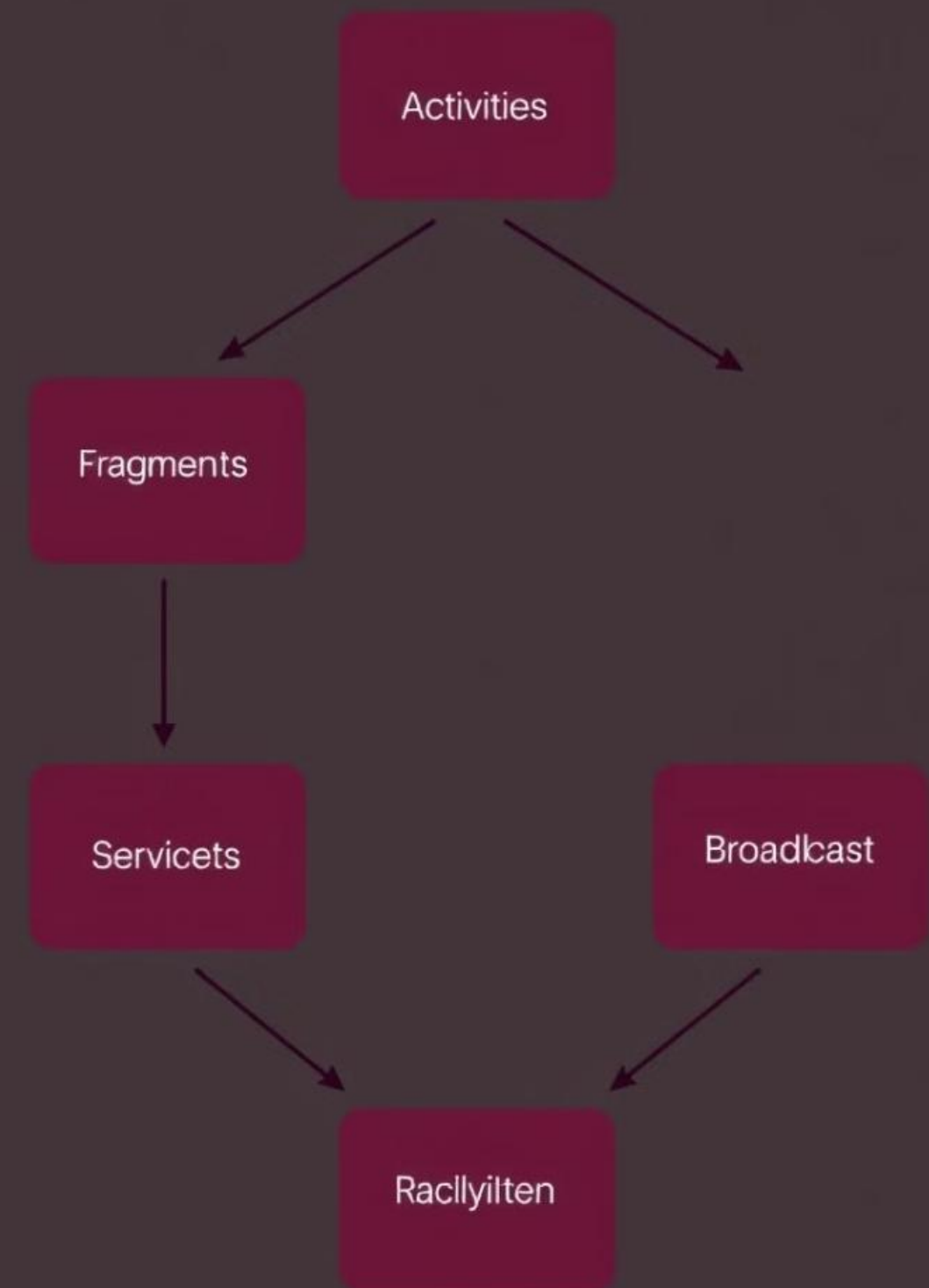
Los Intents son objetos que permiten iniciar otras Activities o enviar información entre componentes.

Broadcast Receivers

Los Broadcast Receivers permiten a los componentes escuchar y responder a eventos del sistema o de otras aplicaciones.

Content Providers

Los Content Providers permiten a las aplicaciones compartir datos con otras aplicaciones.





Servicios y Procesos

Los Servicios se ejecutan en segundo plano y realizan tareas que no requieren una interacción directa con el usuario. Los procesos son las unidades de ejecución que alojan los componentes de la aplicación.



Ejecución en Segundo Plano

Los Servicios pueden ejecutarse en segundo plano incluso si la Activity no está en primer plano.



Gestión de Procesos

Android gestiona los procesos para optimizar el uso de la memoria y la duración de la batería.



Tipos de Servicios

Existen diferentes tipos de Servicios, como los Servicios de inicio, los Servicios ligados y los Servicios de transmisión.

Seguridad y Permisos

Android utiliza un modelo de seguridad basado en permisos para proteger la privacidad y la seguridad de los datos del usuario. Las aplicaciones deben solicitar permisos para acceder a recursos sensibles.

1 Permisos

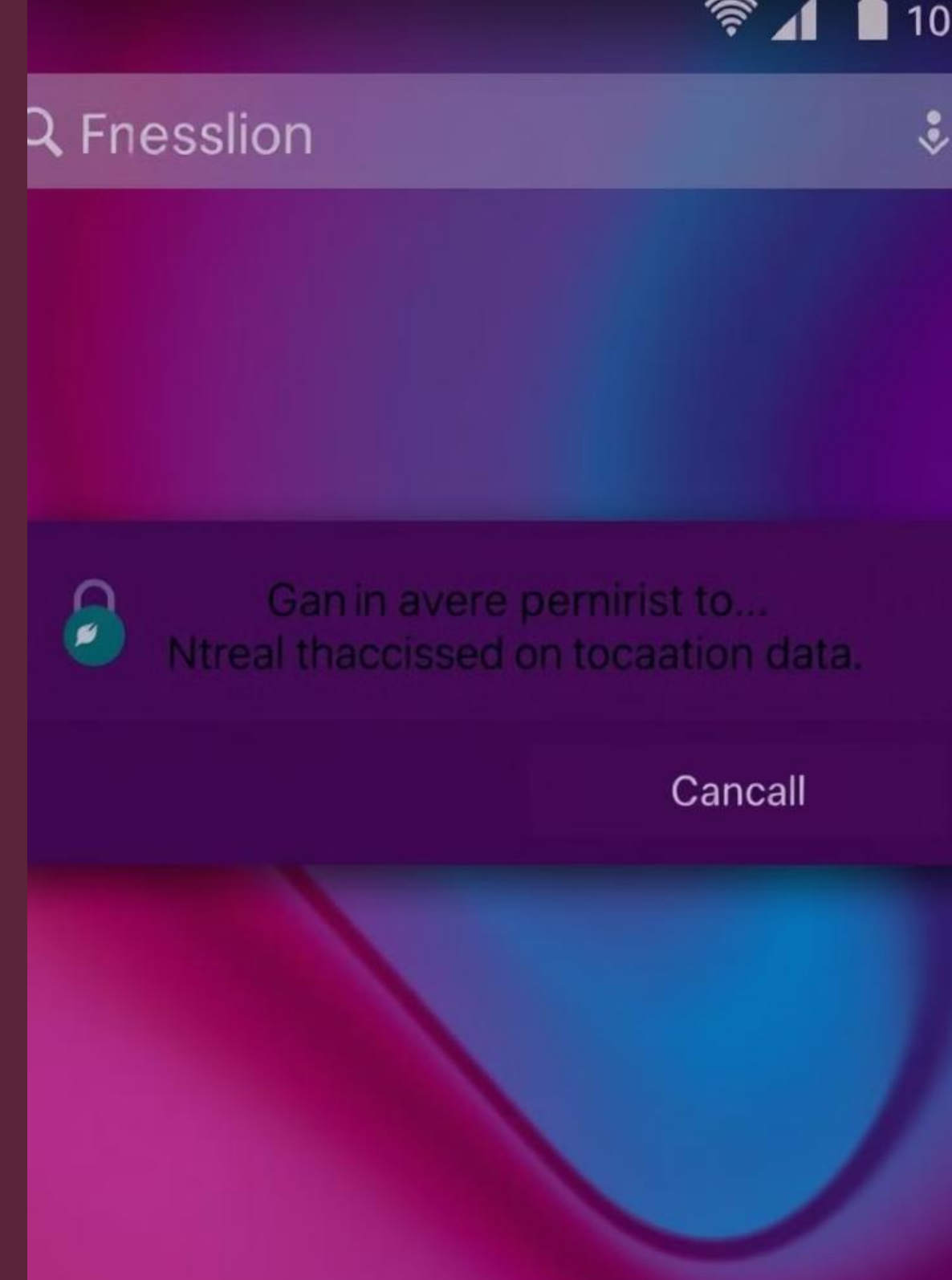
Los permisos son solicitudes que las aplicaciones deben enviar para acceder a datos sensibles como la ubicación, los contactos o la cámara.

2 Modelo de Seguridad

El modelo de seguridad de Android se basa en el principio de "menor privilegio", donde las aplicaciones solo pueden acceder a los recursos que necesitan.

3 Control del Usuario

Los usuarios tienen control sobre los permisos que otorgan a las aplicaciones, lo que les permite gestionar la privacidad de sus datos.



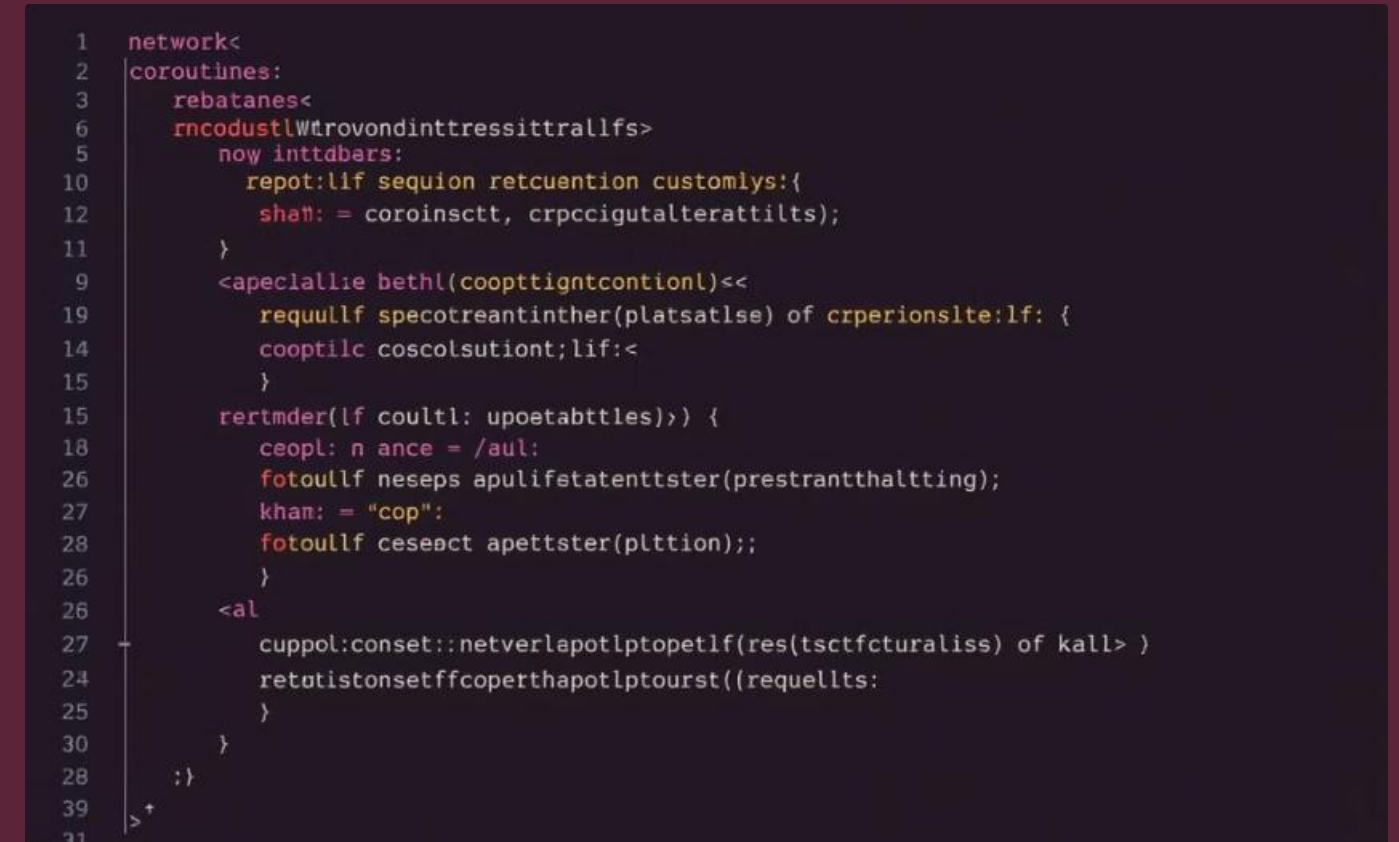
Tendencias y Futuro de la Arquitectura de Android

La arquitectura de Android está en constante evolución para adaptarse a las nuevas tecnologías y las demandas de los usuarios. Las tendencias clave incluyen la modularidad, la seguridad y la mejora de la experiencia del desarrollador.



Jetpack Compose

Jetpack Compose es un nuevo framework para la creación de interfaces de usuario declarativas en Android.



Kotlin

Kotlin se ha convertido en el lenguaje de programación preferido para desarrollar aplicaciones de Android.