



Niveles de API en Android

En este tema, exploraremos el concepto de niveles de API en Android, su importancia para el desarrollo de aplicaciones móviles y las estrategias para optimizar la compatibilidad y rendimiento de las apps.



Presenta: M. en C. Gabriel Hurtado Avilés



Definición de API

Una API (Application Programming Interface) es un conjunto de reglas y especificaciones que permiten la comunicación entre diferentes aplicaciones y componentes de software.

1 Interfaz

Permite a las aplicaciones acceder a funciones y datos de otros componentes de software.

2 Comunicación

Facilita la interacción entre diferentes partes de un sistema.

3 Abstracción

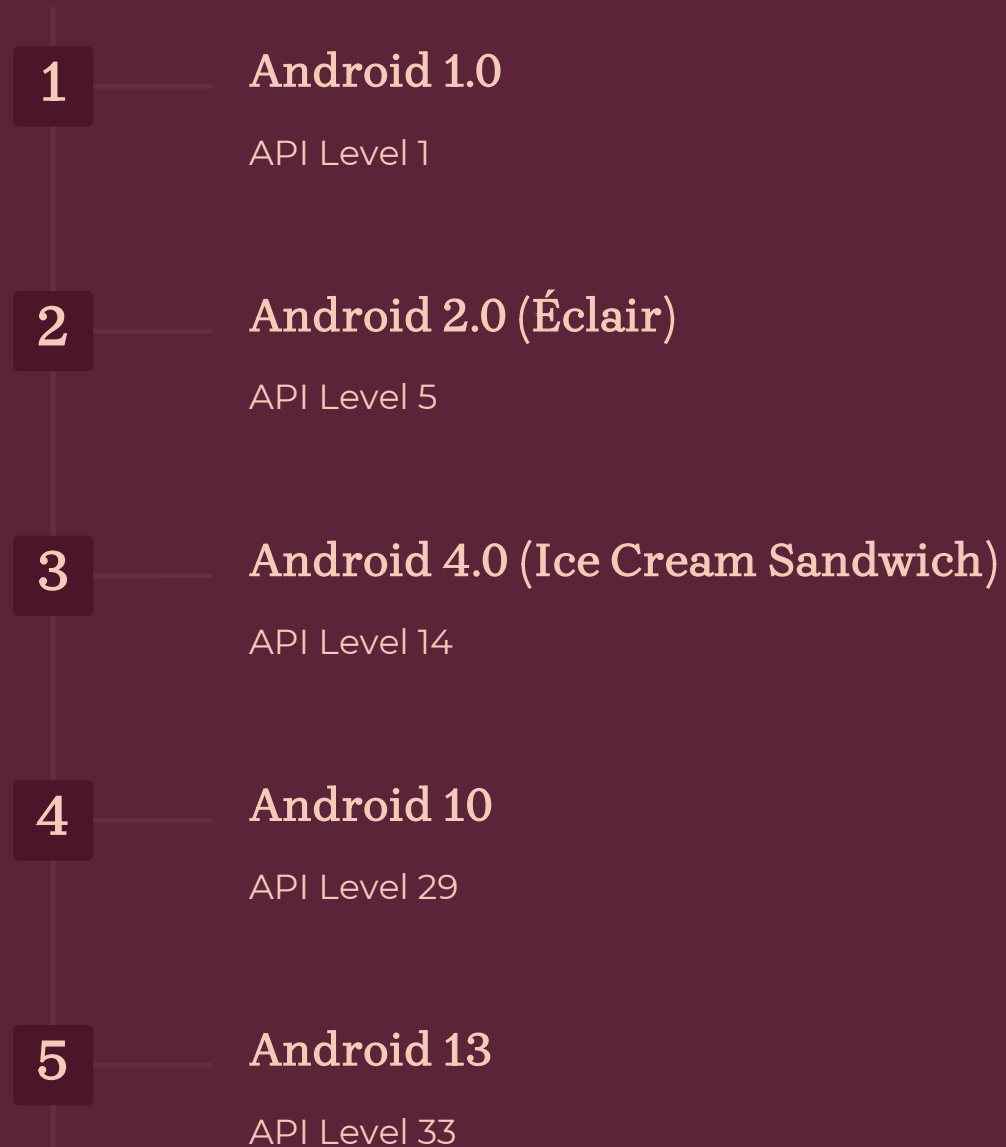
Oculta la complejidad del código y presenta una interfaz simplificada.

4 Reutilización

Promueve la reutilización de código y funcionalidad.

Versiones de Android y Niveles de API

Cada versión de Android (por ejemplo, Android 13) tiene un nivel de API asociado. El nivel de API representa una versión específica del SDK (Software Development Kit).



Importancia de Conocer los Niveles de API

El nivel de API es muy importante para desarrollar aplicaciones compatibles con diferentes versiones de Android. Un nivel de API más alto ofrece acceso a nuevas funcionalidades y características.

Compatibilidad

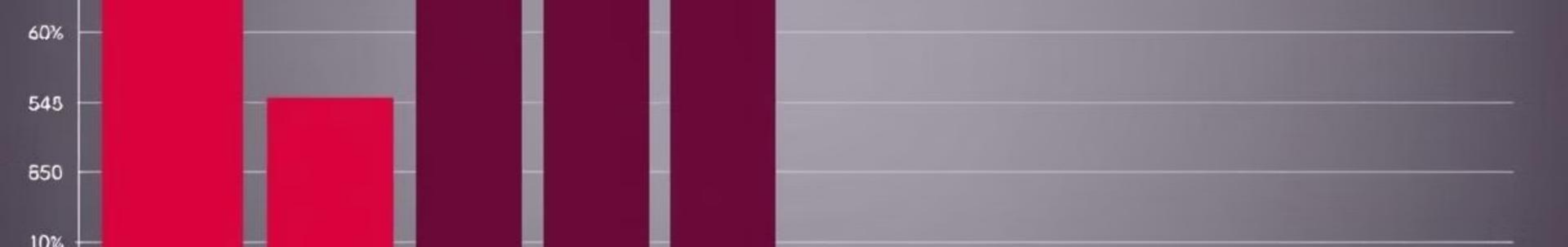
Asegurar que la aplicación funcione correctamente en dispositivos con versiones de Android diferentes.

Funcionalidad

Aprovechar las últimas funciones y características de Android.

Rendimiento

Optimizar la aplicación para aprovechar al máximo los recursos del dispositivo.



Compatibilidad y Fragmentación en Android

Android es un sistema operativo fragmentado, con diferentes dispositivos que ejecutan diferentes versiones de Android. Esto crea un desafío para los desarrolladores.

Nivel de API	Android Version	Porcentaje de dispositivos
API 28	Android 9 Pie	10%
API 29	Android 10	20%
API 30	Android 11	30%
API 31	Android 12	25%
API 32	Android 13	15%

Uso de la API de Android

La API de Android proporciona una colección de clases, interfaces y métodos que permiten a los desarrolladores interactuar con los componentes del sistema operativo.

1

Acceso a sensores

Como GPS, acelerómetro y cámara.

2

Gestión de recursos

Como almacenamiento, memoria y batería.

3

Comunicación de red

Para realizar solicitudes HTTP y manejar conexiones.

4

Interfaz de usuario

Para crear vistas, botones, menús y otros elementos.

	Leve2 API	Leve5 20177	API1 1016 7	Leve2 10315	Leve3 2005.9	Lev 220
Criviesas of Innitlgis cctettiget	U	U	U	U	U	U
APP						
Oniviteaad ectoried	U	U	U	U	U	U
Chasstrice	✓	✓	✓	✓	✓	✓
Oniviesat of thanget cosffer	U	U	U	U	U	U
ArIp	✓	✓	✓	✓	✓	✓
Oniviteandiors and tange	-					
AHFB	✓	✓	✓	✓	✓	✓
Mater						
Dengid praridger	U	U	U	U	U	U
Oniviesad fatured is marker	U	U	U	U	U	U
Statuc	✓	✓	✓	✓	✓	✓
Oniviesad faturad is marker	U	U	U	U	U	U
APN	✓	✓	✓	✓	✓	✓
Onivitead scarker	U	U	U	U	U	U
Pattation	✓	✓	✓	✓	✓	✓
Driviesad mask	U	U	U	U	U	U
Cetrier sesticle	U	U	U	U	U	U
Andiesad fattwad is ceffert	U	U	U	U	U	U
Contion	✓	✓	✓	✓	✓	✓
Endivlean atabic scarer	U	U	U	U	U	U
Oniviesad de lavel sccententr	U	U	U	U	U	U
Onivieead de Inialgie scurger	U	U	U	U	U	U
Ucticerure	U	U	U	U	U	U
Orly iesation tal sket and naidactions	U	U	U	U	U	U
Onivieead de Iniaigic scafert	U	U	U	U	U	U
Andiaties						
Andiesad a cbund scceters	U	U	U	U	U	U

Características y Funcionalidades por Nivel de API

Cada nivel de API trae consigo nuevas características y funcionalidades. Es importante conocer qué funciones son compatibles con diferentes niveles de API.

API 21 (Android 5.0 Lollipop)

Material Design, notificaciones mejoradas y soporte para Bluetooth Low Energy.

API 23 (Android 6.0 Marshmallow)

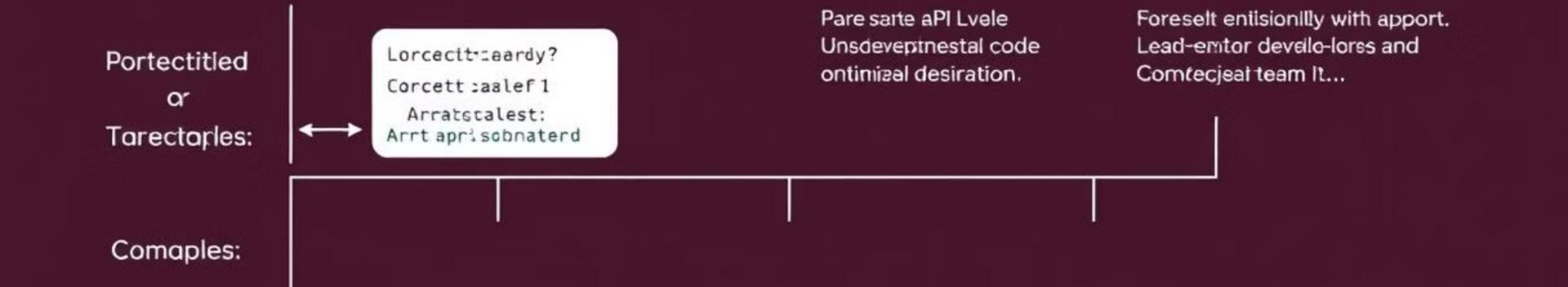
Permisos de runtime, soporte para USB tipo C y Fingerprint API.

API 28 (Android 9 Pie)

Soporte para pantallas plegables, Gesture Navigation y AI Core.

API 30 (Android 11)

Soporte para 5G, burbujas de notificaciones y nuevas API para privacidad.



Estrategias de Desarrollo

Existen diferentes estrategias de desarrollo para manejar la fragmentación de Android. La estrategia dependerá de los requisitos de la aplicación y del público objetivo.



Desarrollo para API mínima

Desarrollar la aplicación para el nivel de API más bajo posible.



Desarrollo para API objetivo

Establecer un nivel de API objetivo para la aplicación.



Desarrollo multiplataforma

Utilizar frameworks multiplataforma para crear aplicaciones compatibles con diferentes versiones de Android.

Porcencia.	tause
user base:	faluiee
features:	feadee

Level :
Piricade = + Worbelicē)
Pertijunce. + }

Consideraciones al Elegir el Nivel de API Objetivo

La elección del nivel de API objetivo es una decisión crucial que influye en la compatibilidad, funcionalidad y rendimiento de la aplicación.

1 Base de usuarios

Considerar la distribución de las versiones de Android en el mercado.

2 Funcionalidad

Determinar las funciones y características necesarias para la aplicación.

3 Rendimiento

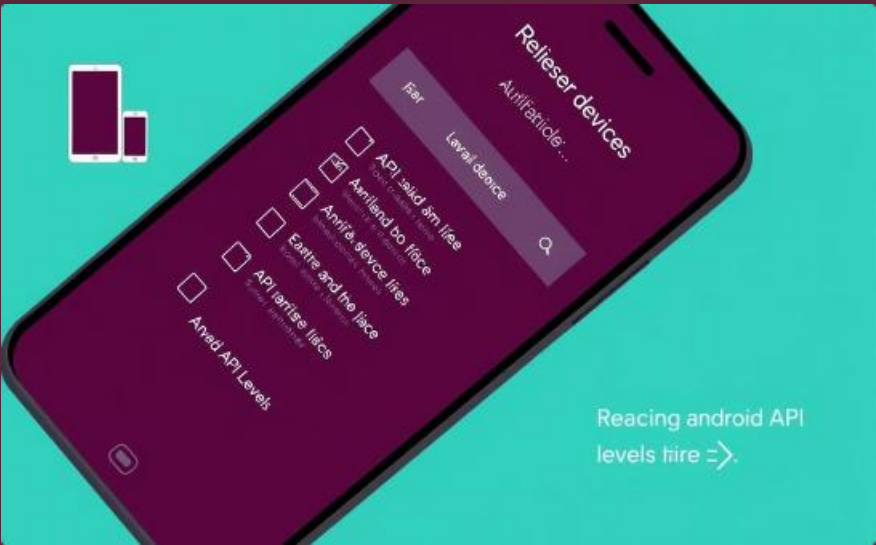
Optimizar la aplicación para un buen rendimiento en dispositivos con diferentes niveles de API.

4 Mantenimiento

Facilitar el mantenimiento y las actualizaciones de la aplicación a futuro.

Conclusiones y Recomendaciones

Conocer los niveles de API es fundamental para el desarrollo de aplicaciones móviles nativas para Android. Al comprender las estrategias de desarrollo.



Compatibilidad

Priorizar la compatibilidad con diferentes versiones de Android.



Rendimiento

Optimizar la aplicación para un rendimiento óptimo en dispositivos con diferentes capacidades.



Actualizaciones

Mantener la aplicación actualizada con las últimas funciones y niveles de API.