

## TEMA 2: A Origem do Android

### Guia Didático com Conteúdos e Exemplos - A Origem do Android

Este guia oferece uma visão detalhada dos conceitos fundamentais relacionados à origem e ao desenvolvimento do Android, incluindo sua história, arquitetura, API, e diferenças entre aplicações nativas e híbridas. O objetivo é fornecer explicações e exemplos claros para facilitar o aprendizado e a preparação para a avaliação.

---

#### 1. História do Android

##### Origem e Evolução

O Android foi criado por uma empresa chamada Android Inc., fundada em 2003 por Andy Rubin, Rich Miner, Nick Sears e Chris White. Inicialmente, o objetivo era desenvolver um sistema operacional para câmeras digitais, mas com a evolução do mercado, a empresa mudou o foco para dispositivos móveis.

Exemplo: Pense no Android como uma plataforma que foi projetada para permitir que smartphones tivessem um sistema operacional robusto e fácil de usar. Sua base em Linux possibilitou maior flexibilidade e personalização.

Em 2005, o Google adquiriu a Android Inc., o que permitiu à plataforma se expandir rapidamente. O Android cresceu tanto que, em 2017, ele superou o Windows como o sistema operacional mais usado para acessar a internet no mundo todo. Esse marco mostrou como os dispositivos móveis (em especial os smartphones) tornaram-se a principal forma de acesso à web.

##### Aquisição pela Google

A aquisição da Android Inc. pela Google em 2005 foi estratégica para a gigante das buscas. O Google viu no Android a oportunidade de criar um ecossistema aberto e expansível para dispositivos móveis, o que levou ao desenvolvimento de uma grande quantidade de dispositivos rodando esse sistema.

Conceito: Ao contrário de sistemas como o iOS da Apple, que é proprietário e fechado, o Android é um sistema de código aberto, o que significa que qualquer desenvolvedor ou fabricante pode modificar o sistema para adaptá-lo às suas necessidades.

---

#### 2. Sistema Operacional Android

##### Baseado no Linux

O Android é construído sobre o kernel do Linux, o que proporciona vantagens como maior segurança, eficiência no gerenciamento de recursos e suporte a diversas arquiteturas de hardware.

Exemplo: O Linux é como a espinha dorsal do Android. Ele gerencia o uso da memória, a execução de processos e a segurança do sistema. Isso permite que o Android execute várias tarefas ao mesmo tempo sem sobrecarregar o dispositivo, mantendo uma boa performance.

Conceito: O kernel do Linux também oferece suporte para hardware de diferentes fabricantes, facilitando o desenvolvimento de smartphones, tablets, e até smartwatches com Android.

### **Código Aberto**

O Android é um sistema de código aberto (open source), o que significa que seu código-fonte está disponível para desenvolvedores do mundo todo. Isso permite que fabricantes de dispositivos, como Samsung e Xiaomi, modifiquem o sistema para seus dispositivos, adicionando funcionalidades e interfaces personalizadas.

Exemplo: Imagine que o Android é como um "kit de ferramentas" que os fabricantes podem usar para construir suas próprias versões. O sistema que você encontra em um Samsung Galaxy pode ser diferente do que está em um Motorola, mas ambos rodam Android com algumas modificações de interface.

---

## **3. Arquitetura do Android**

A arquitetura do Android é organizada em várias camadas que permitem o funcionamento harmonioso de todo o sistema. As camadas principais são:

1. Aplicações: Esta é a camada onde estão localizados os aplicativos que os usuários interagem diretamente, como a agenda, o navegador e o telefone.
2. Framework de Aplicações: Fornece APIs (conjuntos de funções e métodos) que os desenvolvedores usam para criar aplicativos. Isso inclui serviços como o Gerenciador de Atividades e Gerenciador de Janelas.
3. Bibliotecas: Contém bibliotecas C/C++ usadas por diferentes componentes do sistema, como gráficos e banco de dados.
4. Runtime do Android: Inclui a máquina virtual (antes Dalvik, agora ART) que executa os aplicativos Android.
5. Kernel do Linux: O núcleo do sistema, responsável por gerenciar o hardware, incluindo drivers e segurança.

Exemplo: Pense na arquitetura do Android como um edifício. O kernel Linux é a fundação sólida, as bibliotecas e a runtime formam a estrutura interna, o framework de aplicações é como a infraestrutura elétrica e hidráulica, e as aplicações são os móveis e equipamentos que você utiliza no dia a dia.

---

## **4. API (Application Programming Interface)**

### **Conceito de API**

Uma API é um conjunto de funções que permite que programas se comuniquem entre si. No Android, as APIs permitem que os aplicativos acessem recursos do sistema operacional, como a câmera, o GPS ou notificações.

Exemplo: Quando você usa um aplicativo de mapas no Android para encontrar um endereço, o aplicativo utiliza a API de localização do Android para acessar o GPS do dispositivo e mostrar onde você está no mapa.

Importância: As APIs padronizam a forma como os aplicativos interagem com o hardware e outros serviços do dispositivo, tornando o desenvolvimento mais eficiente e compatível com diferentes dispositivos Android.

---

## 5. Aplicações Nativas

As aplicações nativas são programas desenvolvidos especificamente para rodar em um sistema operacional, como o Android, utilizando as ferramentas e APIs oferecidas por esse sistema.

### Vantagens das Aplicações Nativas

Aplicativos nativos têm acesso completo aos recursos do dispositivo, o que permite uma integração mais profunda e um melhor desempenho.

Exemplo: Um jogo desenvolvido nativamente para Android pode tirar melhor proveito do processador gráfico e dos sensores de movimento, oferecendo uma experiência mais fluida e envolvente do que um aplicativo híbrido.

### Ferramentas de Desenvolvimento

Desenvolver aplicativos nativos no Android geralmente envolve o uso do Android Studio, um ambiente de desenvolvimento integrado (IDE), e a linguagem de programação Java ou Kotlin.

---

## 6. Aplicações Híbridas

As aplicações híbridas são construídas usando tecnologias web como HTML, CSS e JavaScript, mas são empacotadas como aplicativos móveis, podendo ser executadas tanto em Android quanto em iOS.

### Vantagens e Desvantagens

- Vantagens: Redução de custos e tempo, já que um único código pode ser utilizado em várias plataformas.
- Desvantagens: As aplicações híbridas, em geral, têm desempenho inferior e menos acesso a recursos nativos do dispositivo.

Exemplo: Aplicações híbridas são como sites adaptados para parecerem aplicativos. Elas podem não ter a mesma performance ou integração profunda com o dispositivo, mas são mais fáceis e rápidas de desenvolver para múltiplos sistemas.

### Frameworks para Aplicações Híbridas

Frameworks como Ionic e React Native são muito utilizados no desenvolvimento de aplicações híbridas, permitindo que desenvolvedores criem uma única base de código para ser usada tanto no Android quanto no iOS.

---

## 7. Frameworks

Um framework é uma estrutura de software que fornece uma base para o desenvolvimento de aplicações, oferecendo bibliotecas e ferramentas que facilitam o processo de criação de software.

### Exemplo de Frameworks no Android

- Android Framework: Fornece as APIs principais que os desenvolvedores usam para interagir com o sistema.
- React Native: Um framework de código aberto que permite o desenvolvimento de aplicações híbridas, utilizando JavaScript para criar interfaces que rodam tanto no Android quanto no iOS.

Conceito: Os frameworks ajudam os desenvolvedores a não "reinventar a roda" cada vez que desenvolvem um aplicativo, fornecendo componentes reutilizáveis e soluções já testadas.

---