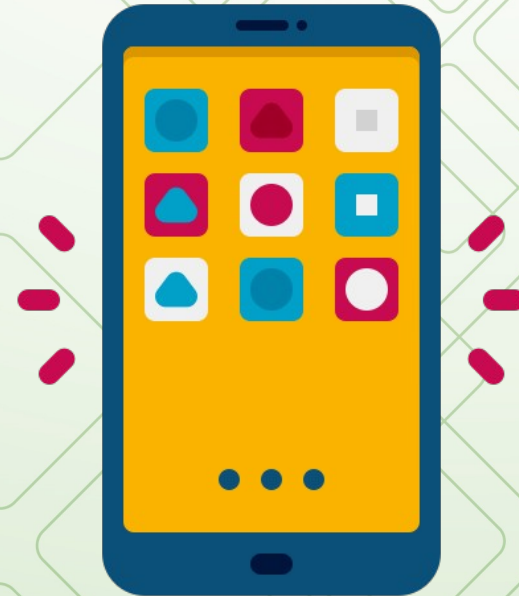
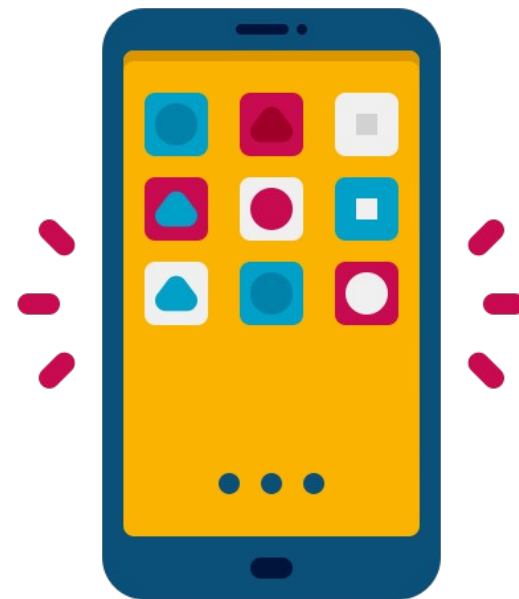


**INSTITUTO
FEDERAL**
Santa Catarina

Programação para Dispositivos Móveis I



Itinerário formativo e plano de ensino



Plano de Ensino

- Programação para Dispositivos Móveis I
- 60 horas
- Terças 19h

Plano de Ensino

Fase	SIGLA	Componente Curricular	CH TEÓRICA	CH PRÁTICA	CH EXTENSÃO	CH EAD	CH TOTAL
5ª Fase	AEX	Atividades de Extensão ²			60		60
	CID I	Ciência de Dados I ³				40	40
	EMP	Empreendedorismo e Inovação ¹	40		20		60
	GSI	Gerenciamento de Serviços para Internet	20	40			60
	PDM I	Programação para Dispositivos Móveis I	10	50			60
Total Carga Horária da 5ª Fase							280
Fase	SIGLA	Componente Curricular	CH TEÓRICA	CH PRÁTICA	CH EXTENSÃO	CH EAD	CH TOTAL
6ª Fase	ESI	Ética, Sociedade e Informática	60				60
	GTI	Gestão e Segurança de Tecnologia da Informação	40	20			60
	CID II	Ciência de Dados II ³				60	60
	PDM II	Programação para Dispositivos Móveis II ¹	10	30	20		60
	TES	Tópicos Especiais em Sistemas para Internet	20	40			60

Ementa Programação para Dispositivos Móveis I

- Habilidades

- Prototipar aplicativos móveis;
- Utilizar ambiente e linguagens de programação para criação de aplicativos móveis;
- Executar aplicações em simuladores e dispositivos reais.

- Conhecimentos

- Tecnologias para desenvolvimento de aplicativos móveis;
- Prototipação de aplicativos móveis;
- Ambiente e ferramentas para construção de aplicações;
- Linguagem de programação para construção de aplicativos móveis;
- Criação de interfaces e programação de eventos;
- Utilização de recursos nativos de dispositivos móveis;
- Execução de aplicações em simuladores e dispositivos reais.

Ementa Programação para Dispositivos Móveis II

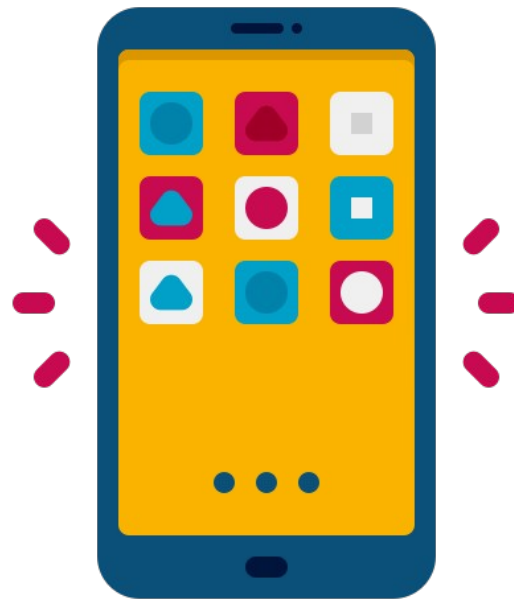
- Habilidades

- Construir aplicações móveis que consomem serviços online;
- Elaborar arquiteturas para soluções que envolvam aplicativos móveis;
- Utilizar mecanismos de segurança em aplicações móveis;
- Empregar tecnologias para construção de aplicações multiplataformas.

- Conhecimentos

- Acesso a serviços online por meio de APIs;
- Arquiteturas de aplicações móveis;
- Acesso a banco de dados local e remoto;
- Segurança em aplicações móveis;
- Tecnologias para construção de aplicações multiplataformas.

Introdução

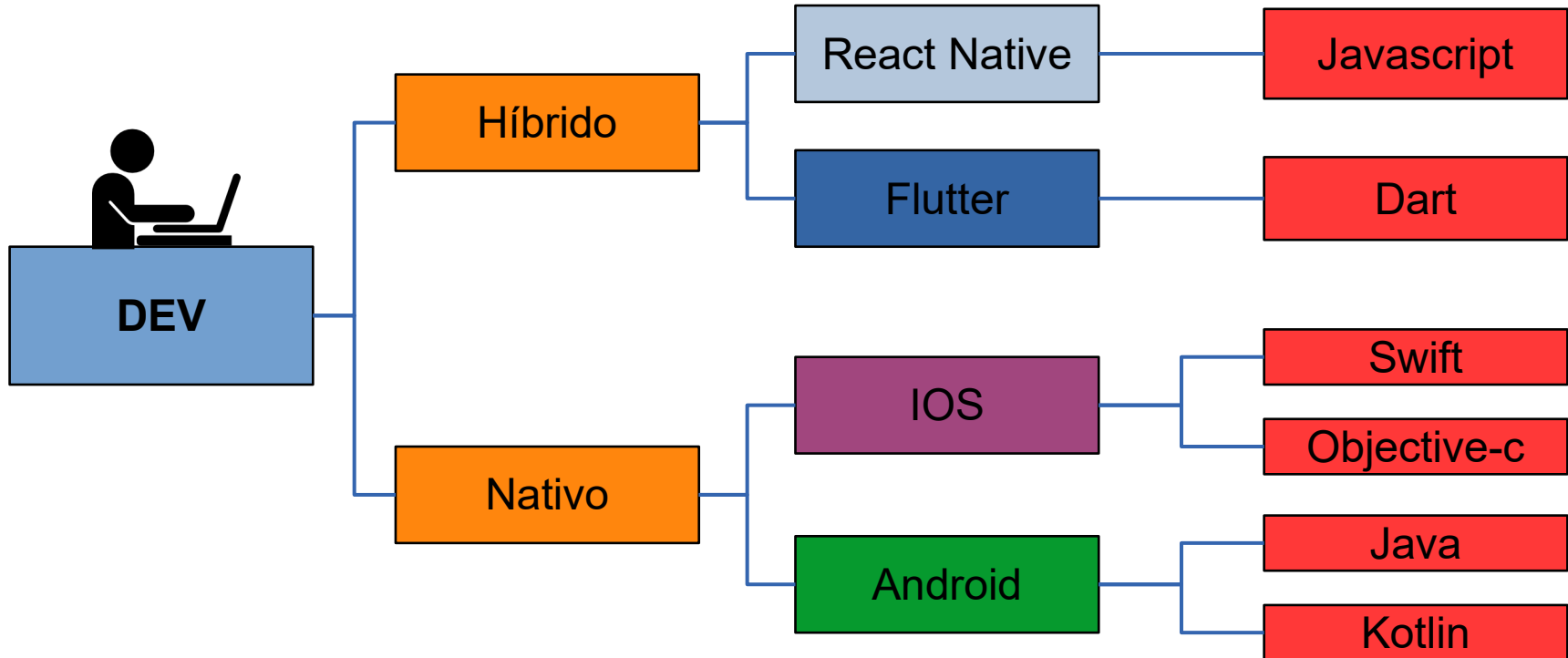


Cenário atual

<https://gs.statcounter.com/os-market-share/mobile/brazil>

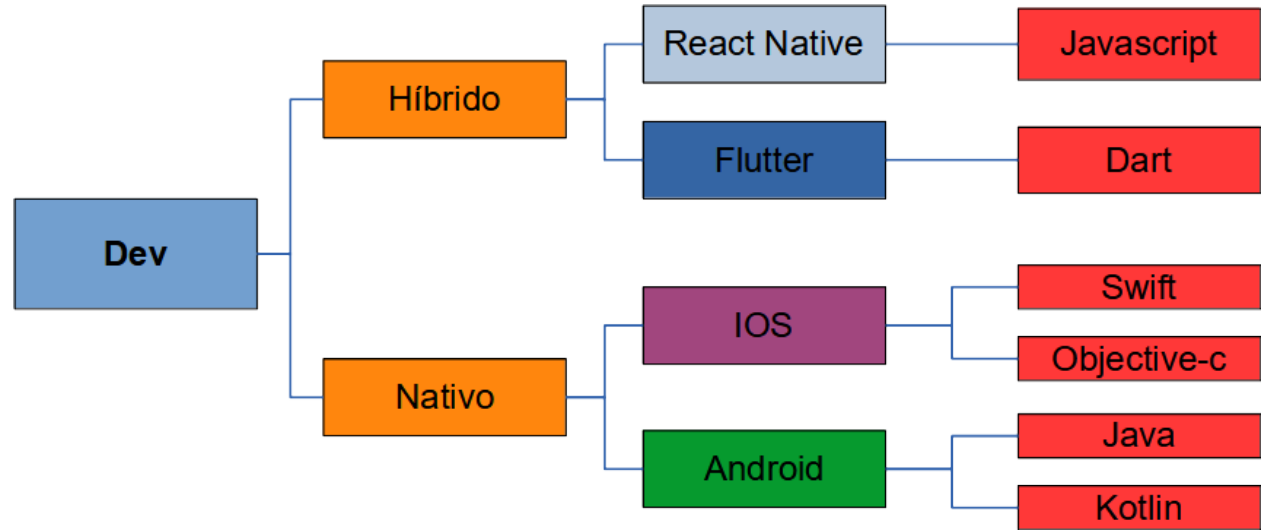


Tecnologias para desenvolvimento de aplicativos móveis



Tecnologias para desenvolvimento de aplicativos móveis

- Quais linguagens de programação vocês conhecem?
- Qual é melhor? Nativa ou Híbrida?



- 100% laboratório informática
- Windows ou Linux
- JDK / JVM
- IDE VsCode / IntelliJ IDEA Community Edition
- Android Studio
- Celular/Cabo USB*

Introdução ao Android



open handset alliance



ANDROID

- Quem fez o Android?
- O Android é o primeiro software lançado pela Open Handset Alliance (OHA), em nov 2007;
- OHA é uma aliança de diversas empresas com a intenção de criar padrões abertos para telefonia móvel;
- Entre as empresas participantes estão Google, HTC, E-Bay, Dell, Lenovo, China Mobile, Intel, Motorola, Qualcomm, Texas Instruments, Samsung, LG, T-Mobile e Nvidia.

<http://www.openhandsetalliance.com/>

Android



- A versão 1.0 do Android chegou em meados de 2008
- O primeiro celular a sair de fábrica com Android foi o T-Mobile G1 (HTC Dream) da HTC
- A partir da versão 1.1, cada lançamento possuía um nome de doce, sendo a primeira 1.5 chamada de Cupcake
- A prática foi abandonada em 2019 com a versão “Android 10”



Cupcake
Android 1.5



Donut
Android 1.6



Eclair
Android 2.0/2.1



Froyo
Android 2.2.x



Gingerbread
Android 2.3.x



Honeycomb
Android 3.x



Ice Cream Sandwich
Android 4.0.x



Jelly Bean
Android 4.1.x



KitKat
Android 4.4.x



Lollipop
Android 5.0



Marshmallow
android 6.0

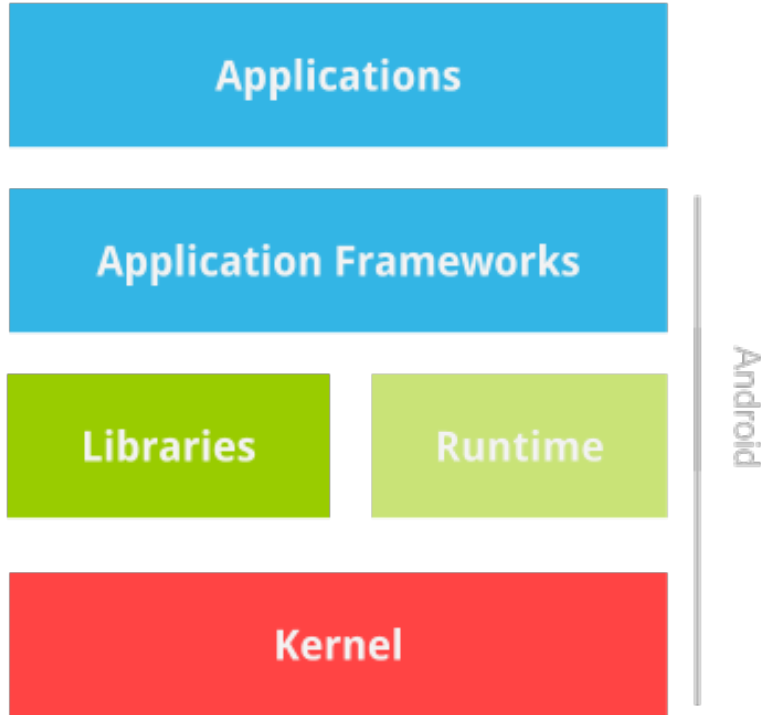


Nougat
android 7.0

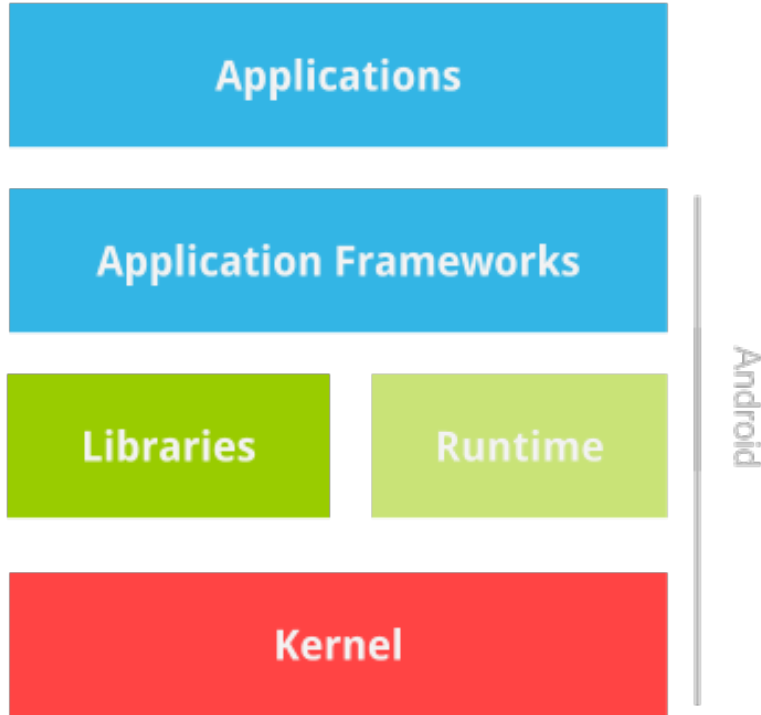


- O Sistema Operacional do Android é baseado no kernel do Linux, responsável por gerenciar a memória, processor, threads, segurança, rede e drivers;
- O Kernel pode decidir encerrar algum processo para liberar recursos e memória, ou até mesmo reiniciar o mesmo processo quando estiver mais ocioso;
- A segurança do Android também é herdada do Linux. Cada aplicativo é executado por um único processo e possui uma thread dedicada;
- Para cada aplicativo instalado é criado também um usuário no sistema operacional para ter acesso a sua estrutura de diretórios. Dessa forma, nenhum app poderá ter acesso a pasta de outra aplicação, ficando completamente isolados.

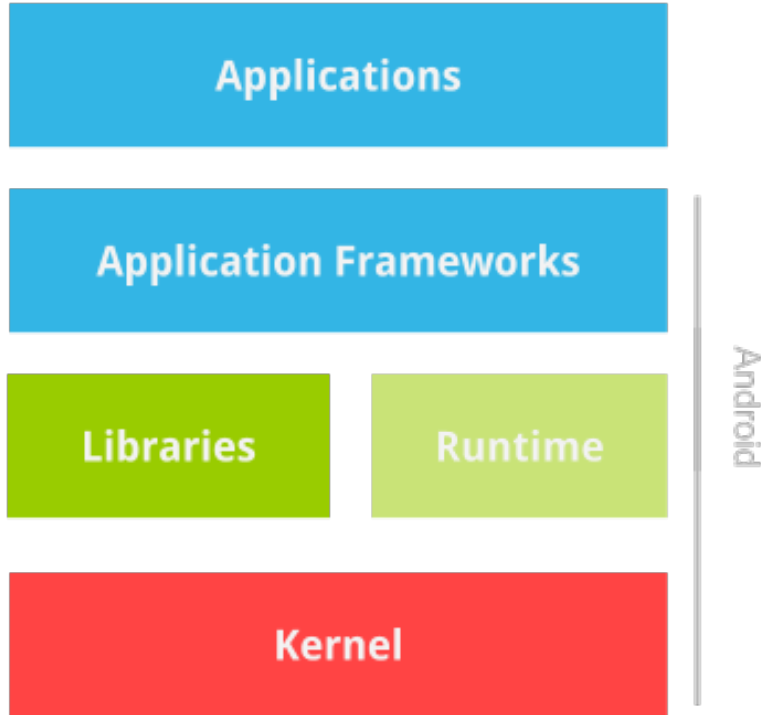




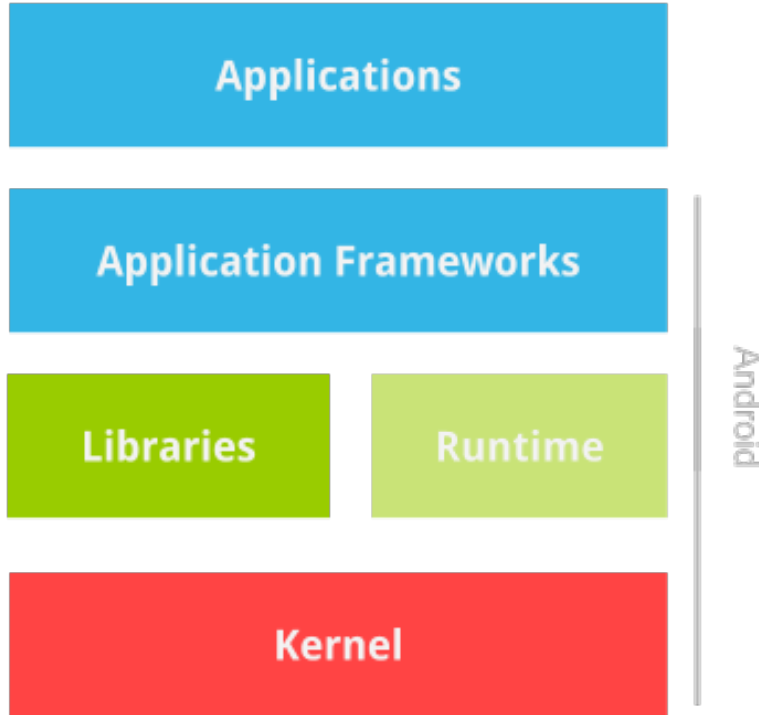
- Camada **Applications**: Onde encontram-se todos aplicativos do Android.
 - Navegadores
 - Mapas,
 - Jogos,
 - Sms
 - Discadores
 - Etc.



- Camada **Application Frameworks**: APIs e recursos utilizados pelos aplicativos.
 - Componentes visuais
 - View system
 - Gerenciador de recursos
 - Eventos que ocorrem
 - Gerenciador de notificações
 - Gerenciador de localizações
 - Etc.



- Camada **Libraries**: conjunto de bibliotecas utilizadas pelo sistema
 - Aceleradores de hardware
 - Renderização 3D
 - Funções de acesso ao banco de dados SQLite
 - Bibliotecas multimídia
 - Etc.



- Camada **Runtime**: Instância da máquina virtual criada para cada aplicação executada.
 - Apesar dos aplicativos possuírem bytecode Java, ele não utiliza uma JVM.
 - Até a versão 5, o Android utilizava a máquina virtual Dalvik, que requer pouca memória e é projetada para permitir que múltiplas instâncias da máquina virtual rodem ao mesmo tempo, deixando para o sistema operacional o isolamento de processo.
 - Dalvik foi substituída pela Android runtime (ART), sendo a principal diferença a compilação em código de máquina na instalação e não na primeira execução.

Discussão

- É possível construir aplicativos para celular em qualquer linguagem de programação?
- Aplicativos podem rodar tanto em IOS quanto Android?
- É necessário antivírus em smartphones?
- Como distribuir aplicativos?

Para saber mais

- 1) Por que novos Sistemas Operacionais se já existe o Android?
- 2) Qual a compatibilidade dos apps atuais?
- 3) Qual o estágio atual de desenvolvimento?
- 4) O Android será substituído em breve?

HarmonyOS

