



Desenvolvimento para Dispositivos Móveis

Introdução

José Pacheco de Almeida Prado
jose.prado@faculdadeimpacta.com.br

Objetivos

- Apresentar os diferentes modelos de desenvolvimento de aplicativos para dispositivos móveis
- Apresentar as diferentes ferramentas e tecnologias existentes para o desenvolvimento de aplicativos para dispositivos móveis
- Apresentar e praticar conceitos e técnicas de desenvolvimento ágil e *desing sprint* para o desenvolvimento de aplicativos para dispositivos móveis
- Capacitar o aluno para o desenvolvimento de aplicativos para plataforma Android

Plano de Ensino - Competências

- Conhecer e entender os modelos de programação para dispositivos móveis: nativo, web e híbrido
- Conhecer as tecnologias disponíveis para desenvolvimento de aplicativos para dispositivos móveis
- Utilizar técnicas de métodos ágeis e *design sprint* para projetar aplicativos para dispositivos móveis
- Conhecer os conceitos fundamentais e a arquitetura para desenvolvimento de aplicações para dispositivos móveis
- Conhecer a arquitetura e os recursos disponíveis na biblioteca Android

Plano de Ensino - Habilidades

- Desenvolver aplicações para dispositivos móveis utilizando os diversos componentes visuais
- Aplicar técnicas de métodos ágeis e *design sprint* no projeto de um aplicativo móvel
- Desenvolver aplicativos móveis híbridos utilizando HTML, CSS e JavaScript
- Desenvolver aplicativos móveis utilizando a plataforma Android e utilizar seus diferentes recursos

Plano de Ensino - Conteúdo

- Conceitos fundamentais, modelos de desenvolvimento de aplicativos móveis, tecnologias
- Uso do *Desing Sprint* para o desenvolvimento de aplicativos móveis: teoria e prática
 - Entendimento/ Definição do problema
 - Divergência
 - Decisão
 - Protótipo
 - Validação

Plano de Ensino - Conteúdo

- Conceitos fundamentais para o desenvolvimento de aplicativos para Android
 - Activity, Intent, Views e Layout
- Recursos disponíveis na biblioteca Android
 - Fragments e Material Design
 - Web services
 - Persistência e banco de dados
 - Recursos Multimídia
 - Services e Jobs
 - Notificações e Push

Plano de Ensino – Bibliografia Básica

- LECHETA, R. R. **Google Android: aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 2.ed. São Paulo: Novatec, 2010.
- LECHETA, R. R. **Google Android 5ª edição: Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 5. ed. São Paulo: Novatec Editora, 2015.
- STARK, J.. **Construindo Aplicativos Android com HTML, CSS e JavaScript**. 1st ed. Sebastopol: O'Reilly/Novatec, 2012.
- STARK, J.. **Building iPhone Apps with HTML, CSS, and JavaScript**. 1st ed. Sebastopol: O'Reilly, 2010.
- GOOGLE INC. **Design Sprint**, 2015. Disponível em:
<<https://developers.google.com/design-sprint/downloads/DesignSprint-NewFormFactors.pdf>>. Acesso em: 6 ago. 2016
- GOOGLE INC. **Design Sprint Methods**, 2015. Disponível em:
<<https://developers.google.com/design-sprint/downloads/DesignSprintMethods.pdf>>. Acesso em: 6 ago. 2016

Plano de Ensino – Bibliografia Básica

- **LECHETA, R. R. Google Android: aprenda a criar aplicações para dispositivos móveis com o Android SDK. 2.ed. São Paulo: Novatec, 2010.**
- LECHETA, R. R. **Google Android 5ª edição: Aprenda a criar aplicações para dispositivos móveis com o Android SDK.** 5. ed. São Paulo: Novatec Editora, 2015.
- STARK, J.. **Construindo Aplicativos Android com HTML, CSS e JavaScript.** 1st ed. Sebastopol: O'Reilly/Novatec, 2012.
- STARK, J.. **Building iPhone Apps with HTML, CSS, and JavaScript.** 1st ed. Sebastopol: O'Reilly, 2010.
- GOOGLE INC. **Design Sprint**, 2015. Disponível em: <<https://developers.google.com/design-sprint/downloads/DesignSprint-NewFormFactors.pdf>>. Acesso em: 6 ago. 2016
- GOOGLE INC. **Design Sprint Methods**, 2015. Disponível em: <<https://developers.google.com/design-sprint/downloads/DesignSprintMethods.pdf>>. Acesso em: 6 ago. 2016

Plano de Ensino – Bibliografia Complementar

- NUDELMAN, G. **Padrões de Projeto para o Android: Soluções de Projetos de Interação para Desenvolvedores**. 1. ed. São Paulo: Novatec Editora, 2013.
- FLING, B.. **Mobile Design and Development: practical concepts and techniques for creating mobile sites and web apps**. 1st ed. Sebastopol: O'Reilly, 2009.
- JUNIOR, M. **Desenvolvendo Sistemas Para Celular**. 1.ed. Rio de Janeiro: Ciência Moderna, 2010.
- FIRTMAN, M. **Programming the Mobile Web**. 1st ed. Sebastopol: O'Reilly, 2010.

Plano de Ensino - Critérios de Avaliação

Nota Final = (30% MAC + 40% Prova + 30% MPAI)

SE (Nota Final $\geq 6,0$ e Frequência $\geq 75\%$) ENTÃO

Aprovado

SENÃO

Reprovado

Em que:

MAC (Média de Atividades Contínuas):

Média das 04 melhores médias de cada AC semanal em um total de 05 ACs.

Prova = Avaliação Semestral

MPAI = Média das provas do PAI para Disciplinas Incidentes do PAI

Evolução da telefonia móvel

- Primeiros telefones móveis tinham uma funcionalidade: fazer ligações.
- Primeiro telefone: Motorola DynaTAC
 - \$4000,00
 - 790 g
 - 25 cm (sem antena)



http://img.timeinc.net/time/photoessays/2010/100_gadgets/communication/motorola_dynatac.jpg

Evolução da telefonia móvel

- Com o tempo, novas funcionalidades foram agregadas:
 - Agenda
 - SMS
 - Despertador
 - Jogos
 - Calculadora
 - Tela colorida
 - ...



Evolução da telefonia móvel

- Hoje vivemos a era dos Smartphones
 - Acesso a internet
 - Tela sensível ao toque
 - Câmera
 - GPS
 - Jogos mais complexos
 - Música
 - Armazenamento de dados
 - E inclusive faz ligação!



Evolução da telefonia móvel

- Smartphones são computadores de mão
 - Os mais modernos tem capacidade de processamento igual ou superior a desktops
- Estas características possibilitam a instalação de programas no telefone móvel.
 - O usuário pode customizar o telefone com os programas ou aplicativos.
- Com isso, abriu-se um novo mercado na área de tecnologia: desenvolvimento de aplicativos para dispositivos móveis.

Porque desenvolver para mobile?

- Auxilia e agiliza seu negócio
 - Integrar aplicativo móvel com sistemas de back-end
- Agrega valor ao seu produto
- Pessoas acessam seu produto de qualquer lugar
 - Mais de 3,2 bilhão de pessoas têm smartphone¹
 - 187 milhões no Brasil²
 - Acesso de qualquer lugar
- Mercado em alta³
 - Estimativa de 529.000 empregos diretos na Europa, sendo 60% em desenvolvimento

¹ <http://www.digitimes.com/news/a20151217PD209.html>

² <http://www.imcgrupo.com/impress/gt/upload/PesTI2016GVcia.pdf>

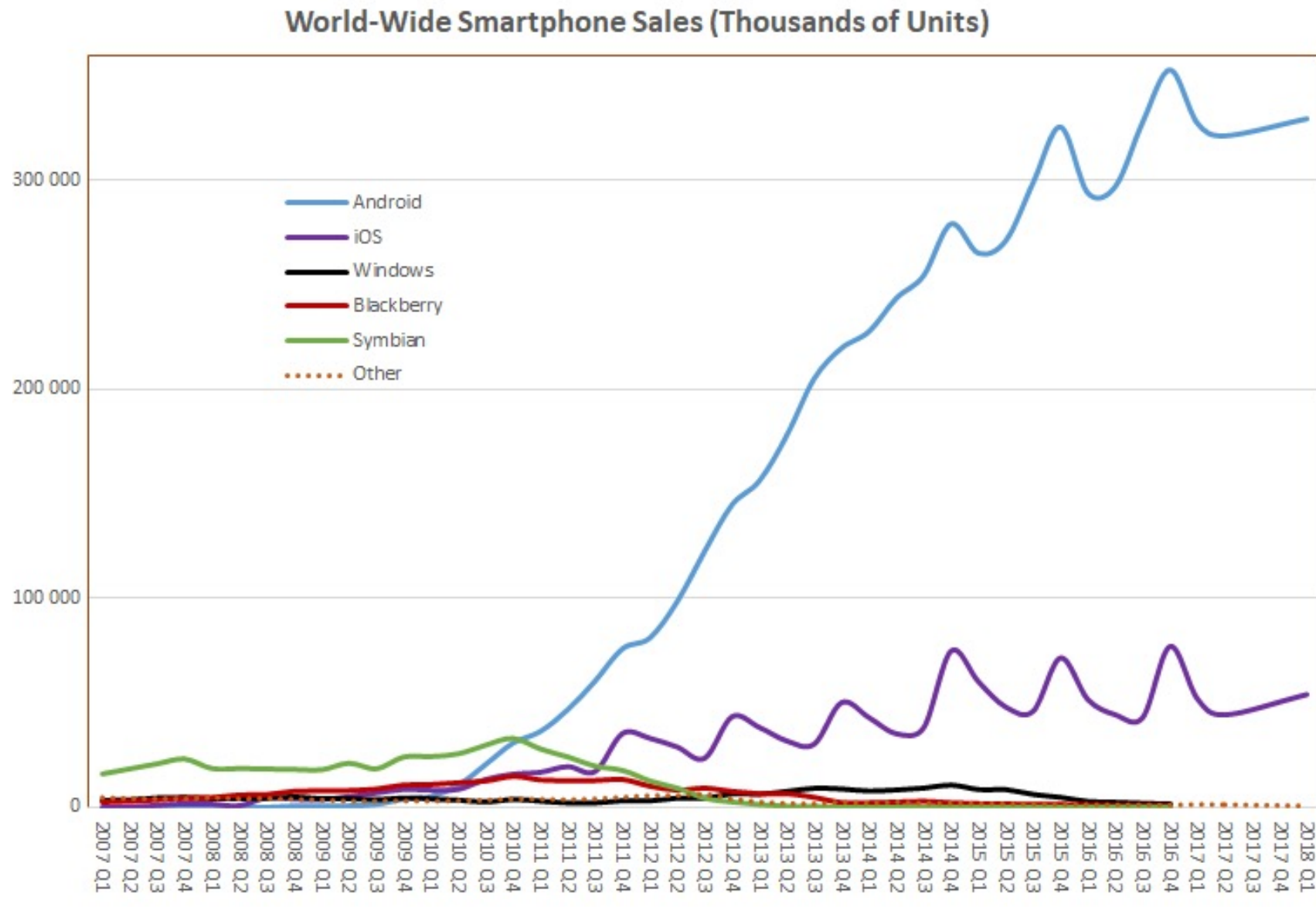
³ <http://www.visionmobile.com/product/the-european-app-economy>

Porque desenvolver para mobile?

- Poucas opções de arquiteturas
 - Quase 99% dos smartphones vendidos no mundo em 2016 utilizam Android e iOS
 - Maior facilidade para manter a portabilidade e disponibilidade de apps

¹ http://en.wikipedia.org/wiki/Mobile_operating_system

Porque desenvolver para mobile?



Fonte: https://en.wikipedia.org/wiki/Mobile_operating_system

Exemplos de aplicativos

- Transações bancárias
- Jogos
- Pagamento por código de barras
- Acesso a dados e serviços
- Redes sociais
- Organizadores pessoais
- Parte de um produto ou serviço
- ...

¹ http://en.wikipedia.org/wiki/Mobile_operating_system

Principais plataformas



iOS

- Sistema Operacional da Apple Inc.
- iPhone, iPad, iPod, Apple TV
- Sistema proprietário
 - Não é licenciado para outros hardwares
 - Não é de código aberto
- Versão atual: 15 (beta)

iOS

- Desenvolvimento (oficial)
 - Objective C ou Swift
 - Necessita de um computador da Apple com:
 - Mac OS X 10.12.6 ou superior
 - Xcode – ambiente de desenvolvimento (gratuito)
 - iOS SDK – biblioteca
 - Registro de desenvolvedor (\$99 / ano)
 - <https://developer.apple.com/support/compare-memberships/>
 - Existe um programa para universidades, mas é limitado

iOS

- Desenvolvimento
 - Os aplicativos desenvolvidos só podem ser testados no simulador, em aparelhos registrados para seu login de desenvolvedor ou enviando para telefones selecionados (utilizando TestFlight Beta Testing)
 - Não é possível instalar o aplicativo em outros dispositivos antes de publicar na App Store
 - Após a submeter o aplicativo, este será avaliado pela Apple, que libera ou não a publicação do aplicativo

iOS

- Passos básicos para desenvolver para iOS
 - Ter um computador Apple (pago)
 - Baixar Xcode e iOS SDK (gratuito)
 - Cadastrar um Apple ID (gratuito)
 - Cadastrar-se como desenvolvedor (pago)
 - Cadastrar e publicar aplicativo (gratuito)

iOS

- Vantagens
 - SO é otimizado para o hardware
 - Rápido para testar no simulador
 - Construtor de interfaces
 - Opções de resolução de tela bem definidas
- Desvantagens
 - Poucas opções de hardware (menos usuários)
 - Sistema proprietário
 - Maior custo para desenvolvimento
 - Apenas uma opção de plataforma de desenvolvimento (para poder publicar e testar)

Android

- Sistema operacional da Open Handset Alliance (<http://www.openhandsetalliance.com>)
 - Liderado pelo Google
 - Motorola, LG, Samsung, Sony, HTC, Toshiba, Telefonica,...
 - Mais de 47 membros, entre provedores, fabricantes de hardware e fabricantes de software
- Baseado em Linux
 - Mesmo kernel de SO como Ubuntu, Debian e outros
 - Android é responsável por gerenciar memória e processos

Android

- Apesar de ser escrito em Java e Kotlin, os apps para Android não utilizam a JVM e os byte codes não são executados.
- As classes Java são compiladas para executáveis da máquina virtual Dalvik
- Dalvik é uma máquina virtual especializada desenvolvida e otimizada especificamente dispositivos móveis.

Android

- Sistema de uso livre e de código aberto (Apache License)
 - Fabricantes customizam o SO
 - Contribui para que o aperfeiçoamento do Android para a plataforma
 - Não é necessário compartilhar as alterações
- Versão atual
 - 11 (Android 11)

https://en.wikipedia.org/wiki/Android_version_history

Code name	Version number(s)	Initial release date	API level
No codename	1.0	September 23, 2008	1
	1.1	February 9, 2009	2
Cupcake	1.5	April 27, 2009	3
Donut	1.6	September 15, 2009	4
Eclair	2.0 – 2.1	October 26, 2009	5 – 7
Froyo	2.2 – 2.2.3	May 20, 2010	8
Gingerbread	2.3 – 2.3.7	December 6, 2010	9 – 10
Honeycomb	3.0 – 3.2.6	February 22, 2011	11 – 13
Ice Cream Sandwich	4.0 – 4.0.4	October 18, 2011	14 – 15
Jelly Bean	4.1 – 4.3.1	July 9, 2012	16 – 18
KitKat	4.4 – 4.4.4	October 31, 2013	19 – 20
Lollipop	5.0 – 5.1.1	November 12, 2014	21 – 22
Marshmallow	6.0 – 6.0.1	October 5, 2015	23
Nougat	7.0 – 7.1.2	August 22, 2016	24 – 25
Oreo	8.0 – 8.1	August 21, 2017	26 – 27
Pie	9.0	August 6, 2018	28
Android 10	10.0	September 3, 2019	29
Android 11	11.0	September 8, 2020	30

Android – Desenvolvimento (oficial)

- Linguagem Java ou Kotlin
- Qualquer computador com
 - JDK
 - Android SDK
 - Android Studio
- Totalmente customizável
- É possível substituir aplicativos nativos
 - Tela inicial, agenda, mensagens...

Android – Desenvolvimento

- Pode-se integrar aplicativos desenvolvidos com qualquer outro aplicativo, de forma muito simples
- Emulador
 - Simula SMS, chamada telefônica, conexão com internet
- Pode-se testar o aplicativo no emulador ou em qualquer dispositivo com Android conectado ao computador
 - Outra opção é copiar o arquivo .apk para o dispositivo
 - O arquivo .apk é gerado após construir o projeto (build)
- Custos
 - Google cobra \$25 de taxa de registro

Android

- Passos básicos para desenvolver para Android
 - Ter qualquer computador
 - Baixar (todos gratuitos):
 - Java JDK (versão 7 8 ou superior)
(<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)
 - Android Studio (já vem com SDK)
(<https://developer.android.com/studio>)

Android

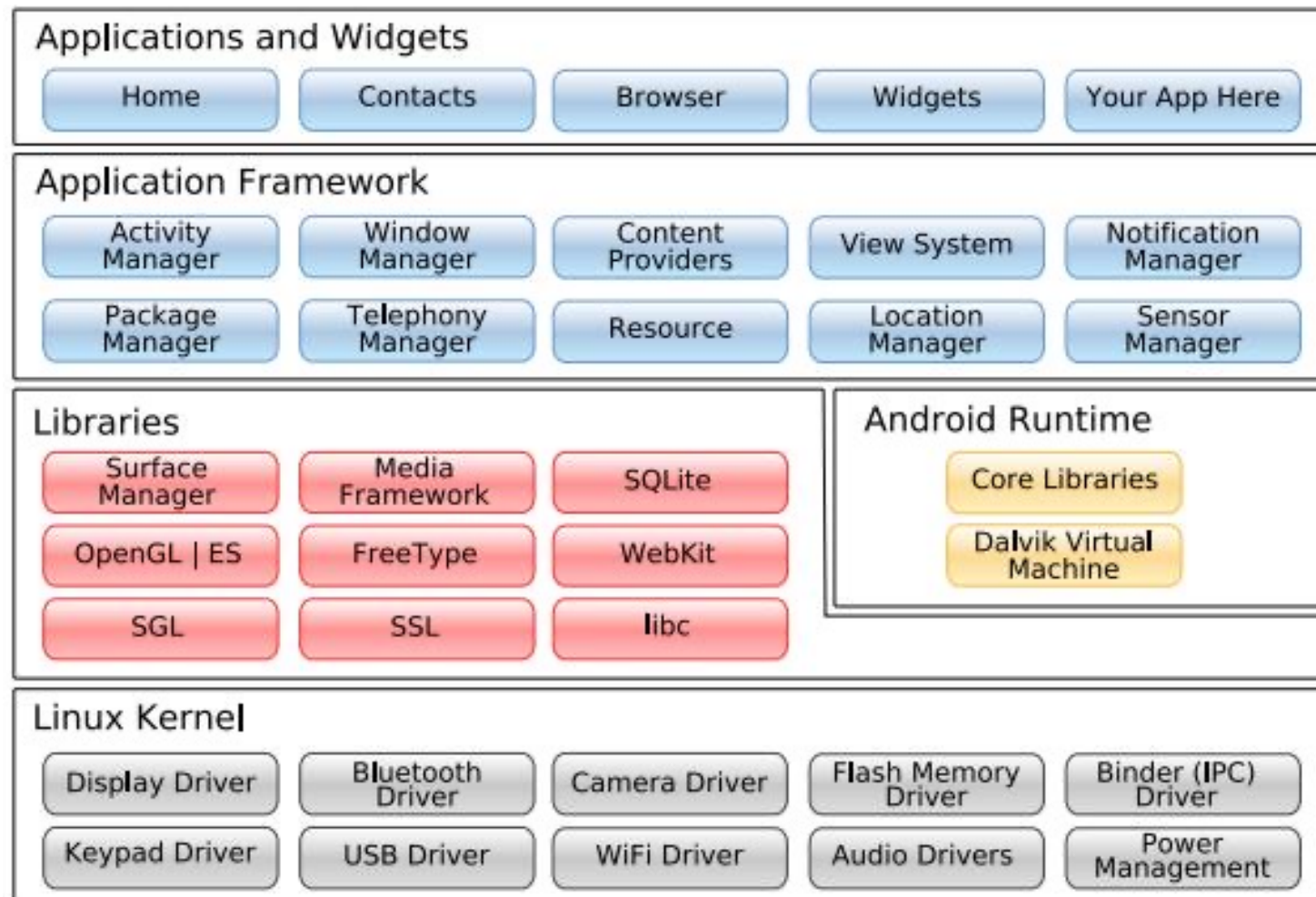
- Passos básicos para desenvolver para Android
 - Registrar uma conta Google (gratuita)
 - Pagar a taxa de desenvolvimento (pago)
 - Publicar aplicativo (gratuito)
 - O aplicativo passa por processo de revisão

Android

- Vantagens
 - Várias opções de hardware (mais usuários)
 - Sistema livre e de código aberto
 - Menor custo para desenvolvimento
 - Desenvolvimento em qualquer plataforma
 - Construtor de interfaces
- Desvantagens
 - SO não é otimizado para o hardware
 - Emulador mais lento
 - Muitas opções de resolução de tela

Android

- Arquitetura



<http://kebomix.wordpress.com/2010/08/17/android-system-architecture/>

iOS x Android

iOS	Android
Buy a Mac, download the free Xcode Installer from the Mac App Store, and start writing code	download the SDK, setup Eclipse and install Google's ADT Plugin
development is done in Objective-C	development is done in Java or C/C++
deploy costs \$99/yr and app has to pass a screening process	Google simple takes a \$25 flat fee to shelf your apps
iOS "Simulator" - runs native code	Android Emulator - runs on a virtual machine
Debug takes 5 seconds on the iOS Simulator	Debug takes about 30 seconds to redeploy and start up in the Emulator on a perfectly-modern machine
has Interface Builder	create UI layouts in XML
Devices have known screen dimensions and hardware	suffers from fragmentation - many versions of the OS and Devices on the market

<http://www.slideshare.net/technologythree/introduction-to-mobiledevelopment#>

Outras formas de desenvolver aplicativos

- Cada plataforma tem sua forma oficial de desenvolvimento
 - Android: Java/Kotlin + Android Studio
 - iOS: Swift + Xcode + MacOS
- Quando o aplicativo é desenvolvido na plataforma oficial, ele é chamado de aplicativo nativo
 - Geralmente melhor desempenho
 - Biblioteca é otimizada para uso do hardware e recursos do hardware
 - Não há camada intermediária entre o aplicativo e a biblioteca do SO

Outras formas de desenvolver aplicativos

- Entretanto, o desenvolvimento nativo não é a única escolha para desenvolver um aplicativo
- Existem basicamente 4 tipos de aplicativos:
 - Aplicativos web
 - Aplicativos híbridos, que agregam características das duas abordagens
 - Aplicativos multiplataforma nativos
 - Aplicativos nativos

Aplicativos Web - características

- Conhecido com Web App
- Acesso pelo navegador ou WebView
- Funciona como um site ou sistema web
 - Normalmente tem uma interface/layout específico para mobile
 - Responsivo
 - Interações e usabilidade diferentes

Aplicativos Web - características

- Desenvolvimento é mais rápido
 - Utiliza tecnologias como HTML, JS e CSS
 - Muito recurso humano disponível
 - A lógica o back-end ou serviço pode ser desenvolvido na linguagem mais apropriada para a equipe
- Recursos de hardware limitado
 - Não acessa tudo que o dispositivo oferece
- Não é acessível pela loja de aplicativos
- Totalmente dependente da internet

Aplicativos Web

- Usuários visualizam mais rapidamente
 - É mais fácil abrir seu app no browser do que pagar para baixar um aplicativo
 - Por estar na web, também é mais fácil de atrair novos usuários fora da plataforma móvel
- Publicação mais rápida
 - Para disponibilizar nova versão basta atualizar os arquivos para o servidor
- Atualização automática
 - Usuário não precisa atualizar o aplicativo

Aplicativos Web

- Globalização
 - Um mesmo projeto pode servir para várias plataformas e dispositivos
 - Android, iOS, Windows, Web...
 - Ou então pode-se aproveitar pelo menos o back-end, onde estará a maior parte das regras de negócio
 - O acesso direto ao servidor também economiza energia do dispositivo
 - Mas acaba gastando mais do plano de dados
- Grande parte da lógica e navegação é executada e um servidor web

Aplicativos Web “Embutido”

- Alguns aplicativos são desenvolvidos como um Web App mas acessados de um aplicativo nativo
- O aplicativo nativo é criado apenas para acessar o Web App em uma web view (assim como faz o navegador) e poder disponibilizar em uma loja de aplicativos
- Continua com limitação de uso dos recursos do hardware
- Continua com dependência da internet

Aplicativos Híbridos

- Desenvolvimento que utiliza tecnologias da web (HTML, CSS e JavaScript), mas que executa dentro de um aplicativo nativo
 - A interface é desenvolvida utilizando HTML, CSS e JavaScript
 - A lógica é feita com JavaScript
 - Não é possível utilizar outra linguagem
 - Apenas tecnologias de cliente web são utilizadas

Aplicativos Híbridos

- O que foi criado utilizando HTML, CSS e Javascript é colocado em um projeto Nativo
 - possibilita acesso a recursos do dispositivo
 - GPS, câmera, contatos, push, notificações
 - O que foi criado com HTML é executado em uma WebView
 - A diferença para um Web App “embutido” é que a parte web está executando no aplicativo, e não em um servidor externo

Aplicativos Híbridos

- Depende de uma biblioteca de terceiros (Cordova, Phonegap)
 - Camada colocada entre o aplicativo e a biblioteca da plataforma
- Desenvolvimento mais rápido, já que utiliza tecnologia web
 - Precisa apenas de um rápido aprendizado para aprender como acessar os recursos nativos pela biblioteca auxiliar
- Possibilita que o aplicativo funcione offline

Aplicativos Híbridos

- Possibilita o desenvolvimento de aplicativos multiplataforma
 - Qualquer plataforma tem um recurso de WebView
- Como é multiplataforma, atinge um maior número e usuários
- Pode publicado na loja e instalado no dispositivo
- Pode ter perda de desempenho para aplicativos muito complexos
- A maior parte das características de um aplicativo nativo estão presentes nos híbridos

Aplicativos Híbridos

- Principais bibliotecas e ferramentas
 - Cordova
 - Phonegap
 - Ionic
 - Appcelerator
 - JQueryMobile
 - Intel XDK

Aplicativos Multiplataforma nativos

- Não utiliza a linguagem da plataforma para desenvolvimento, mas também não executa em uma WebView
- Os componentes são nativos, não HTML
- O funcionamento pode ser diverso:
 - Compilação para a linguagem nativa (compilação cruzada)
 - Biblioteca que acessa direto a biblioteca da plataforma

Aplicativos Multiplataforma nativos

- Possuí a maioria das características de um aplicativo híbrido ou nativo:
 - Multiplataforma (apenas híbrido)
 - Acesso a recursos do dispositivo
 - Funcionamento offline
 - Publicação na loja
 - Instalação no dispositivo

Aplicativos Multiplataforma nativos

- O desempenho depende muito da solução que for utilizar
 - Usualmente uma tecnologia que utiliza compilação cruzada tende a ter menor desempenho, uma vez que não temos controle do que é feito na compilação
- Principais tecnologias
 - Xamarin: desenvolvimento utilizando C#
 - Nativescript: desenvolvimento utilizando Javascript, Typescript e XML
 - React Native: desenvolvimento utilizando Javascript, Typescript e XML

Aplicativos Nativos

- Documentação mais completa
- Maior facilidade para reportar erros
- Bibliotecas próprias
 - Recursos como orientação da tela e multi-toque são suportados pelo navegador (WebKit) no Web App
 - Porém Agenda, GPS, acelerômetro estão apenas em aplicativos nativos

Aplicativos Nativos

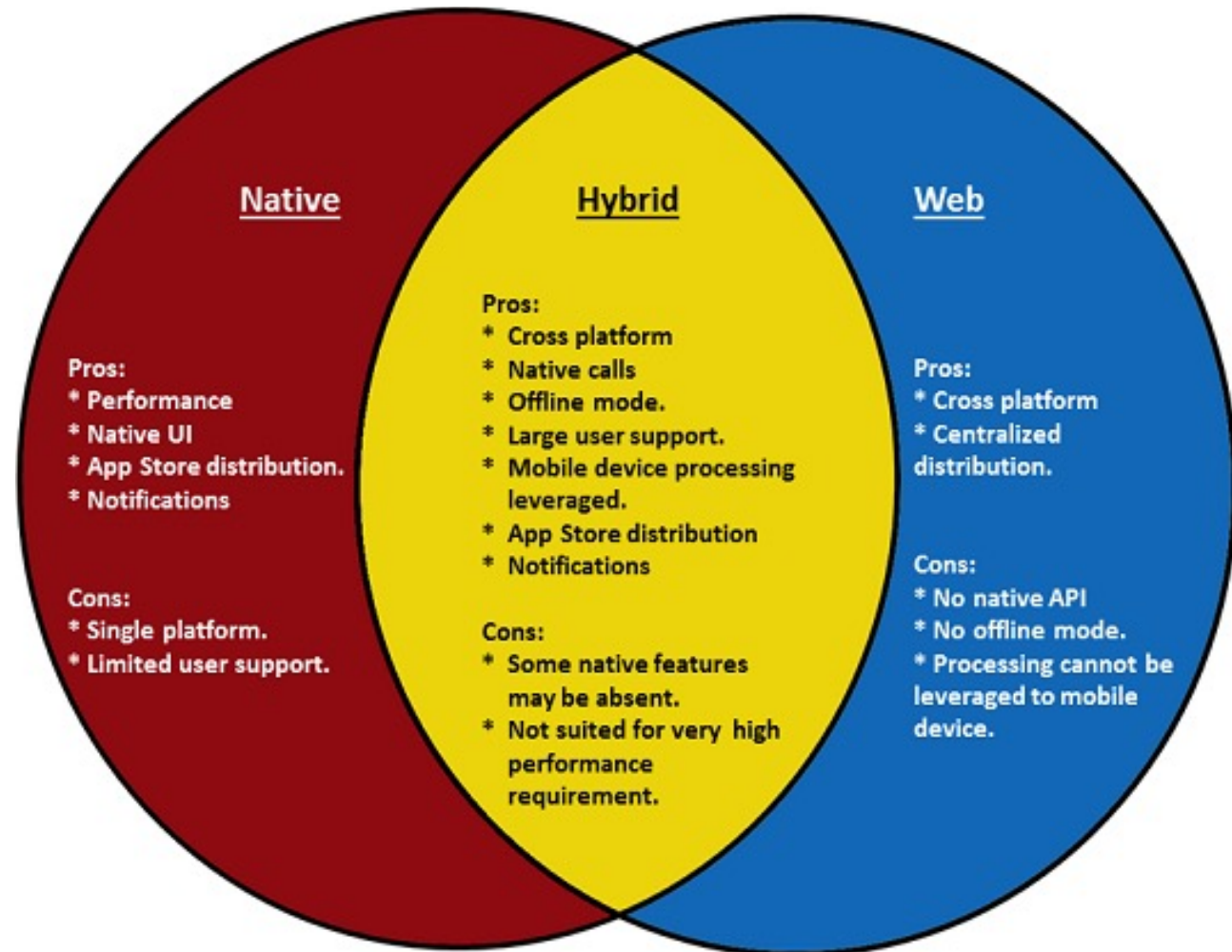
- Disponível na loja de aplicativos
- Instalação no dispositivo
- Criação de um ícone de acesso rápido
- Funcionamento offline
- Velocidade nativa do dispositivo
- Maior segurança

Aplicativos Nativos

- Grande parte das regras de negócio e navegação é executada no dispositivo
- Funciona como um software desktop, com possibilidade de uma arquitetura cliente/servidor (consumo de serviços)
- Armazenamento de dados localmente
- Tarefas em segundo plano

Estratégias de desenvolvimento

- Comparação



http://developer.nokia.com/community/wiki/Arquitetura_de_aplica%C3%A7%C3%B5es_multiplataforma_para_dispositivos_m%C3%B3veis

Estratégias de desenvolvimento

- Porque escolher um aplicativo nativo?
 - Pode funcionar sem a internet
 - Melhor performance
 - Mais fácil de encontrar
 - Armazenamento de dados locais
 - Utilizar recursos do dispositivo
 - Executar tarefas em segundo plano (notificações)
 - Melhor usabilidade e navegação
 - Independente da interpretação do browser de HTML/JS/CSS
 - Boa alternativa quando:
 - Aplicativo é bastante sofisticado
 - Precisa utilizar recursos do dispositivo

Estratégias de desenvolvimento

- Quando escolher um aplicativo nativa?
 - Jogos interativos
 - Uso regular
 - Cálculos complexos
 - Armazenamento de dados
 - Uso de recursos do dispositivo
 - Não precisa de conexão com a internet

Estratégias de desenvolvimento

- Porque escolher um aplicativo web?
 - Pode ser o suficiente para algumas aplicações
 - Desenvolvimento em HTML5, JS e CSS
 - Roda na maioria dos browsers modernos
 - Fácil publicação e manutenção
 - Boa alternativa quando:
 - Aplicativos que não usam recursos no dispositivo
 - Muitas atualizações
 - É totalmente dependente da internet



Obrigado!

