

Primeiros Passos - Mobile

por Gustavo Silva

Por que desenvolver mobile?

- **Bom para empreendedores:**

- Milhões de aparelhos iOS/Android e potenciais usuários;
- Vende no mercado global, mais resistente a crises;
- É pago em dólar (4x mais).

- **Bom para desenvolvedores:**

- Mão de obra de qualidade extremamente escassa;
- Salários bem altos.

- **Bom para freelancers:**

- “Conhece alguém que mexe com aplicativos? Meu amigo/primo/vizinho estava precisando.”

Aplicativos Nativos

- **São os aplicativos feitos na linguagem nativa do aparelho:**

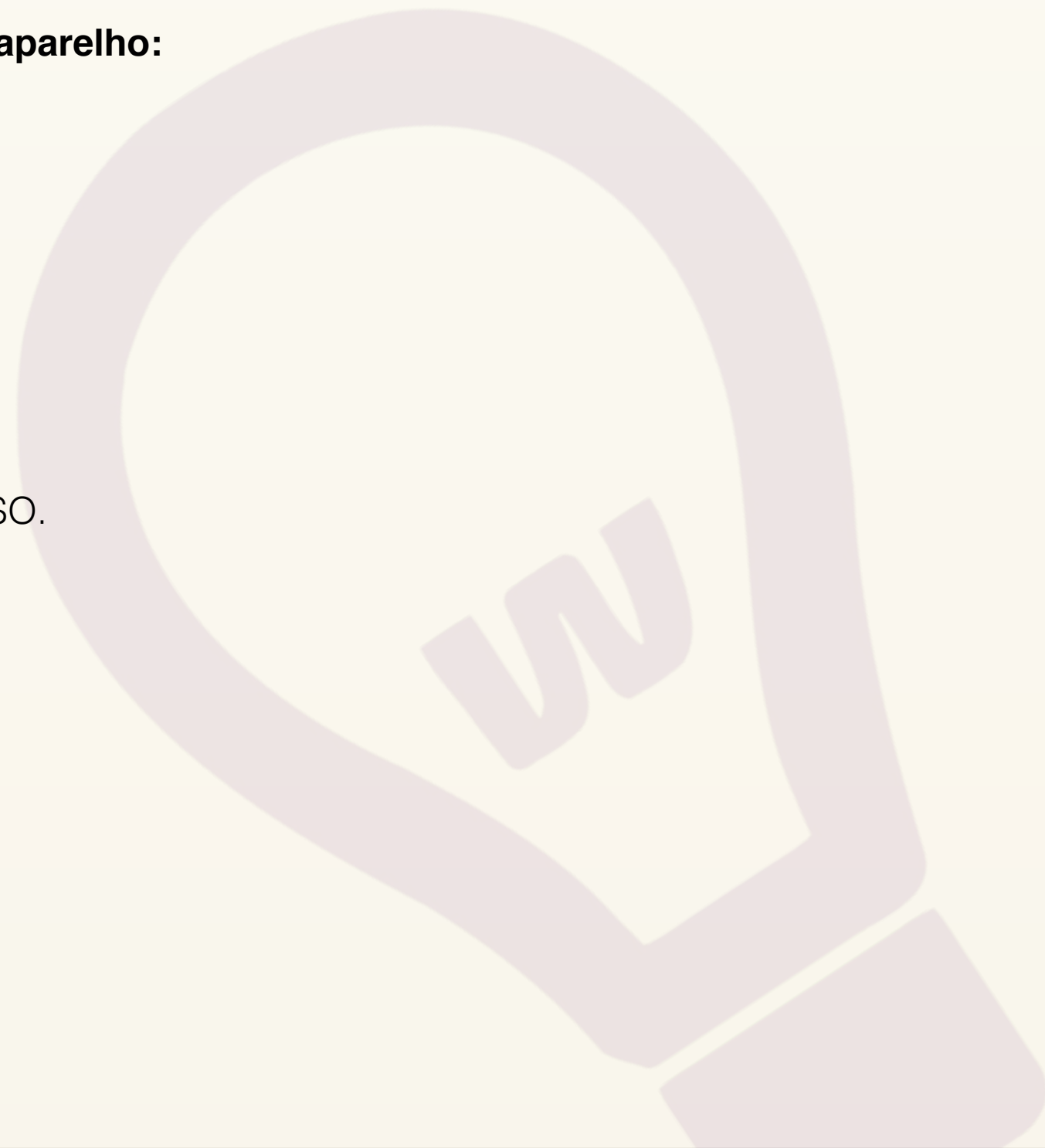
- Objective-C ou Swift para iOS;
- Java para Android.

- **Vantagens:**

- Melhor desempenho;
- Maior aproveitamento do hardware do aparelho.
- Aplicativo mais fluido, agradável e integrado ao SO.

- **Desvantagens:**

- Difícil e caro de aprender;
- Desenvolvimento demora mais;
- Feito para uma plataforma específica;
- Poucos bons profissionais no mercado;
- Ou você é bom em iOS ou Android;
- Se for iOS, vai desembolsar mais de 10k;



Sistemas Web

- **São aplicativos que rodam no browser, relacionados a um servidor. Basicamente são sites com mais funcionalidades.**
- **Vantagens:**
 - Roda em qualquer plataforma (até desktop, televisão, microondas);
 - Bem simples e rápido de desenvolver;
 - Não precisa de licenças de desenvolvedor.
- **Desvantagens:**
 - Má integração com o sistema;
 - Bastante limitado;
 - Roda dentro do browser;
 - Desempenho ruim;
 - Baixa capacidade offline;
 - Pode ser derrubado pelo SO para poupar memória;

E agora?

- **Como conciliar o bom desempenho e integração do aplicativo nativo com a flexibilidade e facilidade de um sistema web?**
 - **A resposta: aplicativos híbridos!**
 - **Vários frameworks: Appcelerator, Cordova, Xamarin.**

O que é Cordova?

- **Breve história:**

- Feito por uma startup chamada Nitobi;
- Chamava-se Phonegap;
- Adobe comprou a Nitobi em 2011 e doou o código para a Apache Foundation. Nascia o Apache Cordova.

- **Cordova é uma plataforma que permite construir aplicativos utilizando tecnologias Web como HTML, CSS e JS, compilados para iOS, Android, Windows Phone, Blackberry e o que mais você quiser.**

- Na verdade trata-se de um aplicativo “wrapper” cuja única tela é um browser, apontando para um site que na verdade está dentro do próprio aplicativo;
- Não há compilação de código HTML para nativo;
- Possui uma série de plugins para acessar funcionalidades do SO e do hardware de dentro do código HTML/JS.

O que é Cordova?

- **Vantagens:**

- Desenvolvimento rápido, fácil e bonito (bom para freelas);
- Desempenho quase nativo;
- Acesso a quase todas as funções do hardware;
- Muitas vezes nem dá pra saber que é híbrido.

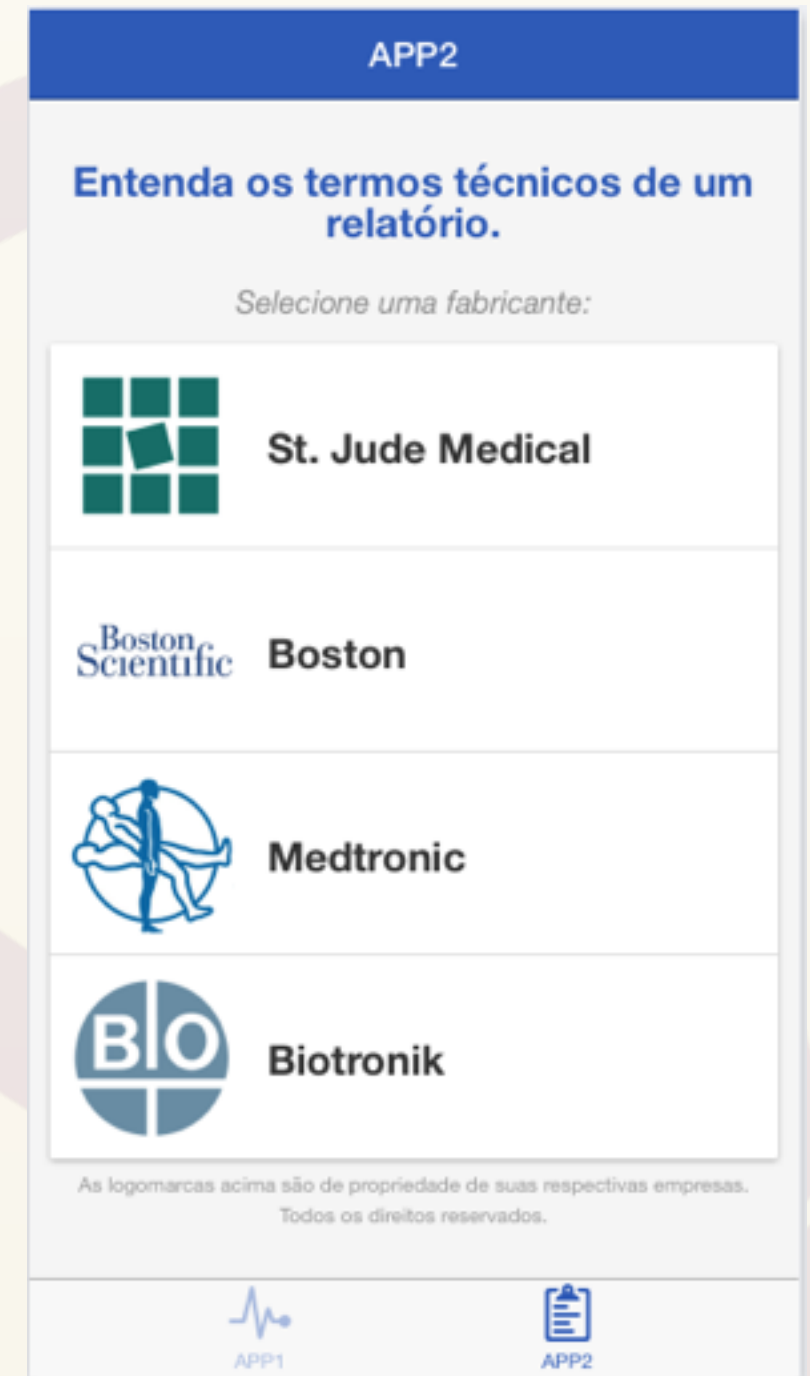
- **Desvantagens:**

- Não é funcional para aplicativos pesados como jogos;
- Não é tão seguro pois o código não é compilado;
- Pode ficar lerdo em celulares mais velhos;
- Pode ficar feio e não-nativo dependendo do cuidado.



Ionic

- **Biblioteca construída sobre o Cordova;**
 - Equivalente ao Bootstrap. Facilita muito a criação de telas bonitas;
 - Além disso, gerencia todo o sistema de paginação do aplicativo.
 - Baseado em AngularJS.
- **Equipe ativa que está o tempo todo lançando novas ferramentas para facilitar a vida do desenvolvedor: Push, Analytics, Deploy, Creator**



NPM

- **Gerenciador de pacotes do Node.js (Node Package Manager);**
- **Boa parte das ferramentas estão disponíveis no NPM;**
- **Pode ser baixado no site do Node.js.**
- **Utilização:** `npm install -g ferramenta`
- **`npm install -g cordova`**
- **`npm install -g ionic`**

Outras Ferramentas

- **Bower:** Gerenciador de bibliotecas do Node.js e do Angular.
- **Gulp/Grunt:** Automatizador de processos de *build* de projetos Javascript, como minificar arquivos.
- **Sass/Less:** Compiladores de CSS, que entendem as funções deste permitindo coisas como variáveis.
- **Cordova CLI:** Ferramenta que administra as funcionalidades do Cordova.
- **Ionic CLI:** Além de englobar as funções do Cordova CLI, possui funcionalidades próprias do Ionic.

Cordova CLI

- **cordova start:** Cria um novo projeto Cordova.
- **cordova platform add:** Adiciona uma nova plataforma ao projeto. Necessário ter o SDK da plataforma instalado.
- **cordova plugin add:** Adiciona um plugin ao projeto. A lista de plugins e seu funcionamento podem ser encontrados na internet.
- **cordova build android:** Compila o projeto e gera um arquivo executável para a plataforma desejada, para fins de debug. Para gerar o arquivo final, utilizar a flag `—release`.
- **cordova run ios:** Compila o projeto e roda em um dispositivo conectado ou em um emulador.

Anatomia de um Projeto

- **platforms:** Pasta com as plataformas de um projeto. Lá estarão os arquivos gerados no build (após escavar um pouco).
- **plugins:** Pasta com os plugins. Não deve ser mexida diretamente.
- **scss:** Pasta com os arquivos Less/Sass que serão compilados.
- **hooks, merges:** Pastas avançadas do sistema.
- **www:** Pasta principal, dentro dela está todo o código, como se fosse um site comum.
- **config.xml:** Arquivo de configuração, onde podem ser mudados o nome do projeto, o ID, a versão atual, além de alguns outros parâmetros que podem ser requisitados por plugins.

Plugins

- **Permitem controlar funções nativas do hardware de dentro do código Javascript;**
- **São acessados por Javascript puro e relativamente complicado;**
- **Para isso, existe a biblioteca ngCordova (dos mesmos criadores do Ionic) que fornece um wrapper Angular para os plugins.**
- **Para buscar plugins: <https://cordova.apache.org/plugins/>**

Plugins

- **Plugins mais famosos:**
 - **Camera:** Acessa câmera (obviamente) e o rolo de fotos.
 - **Statusbar:** Permite modificar a statusbar do aparelho.
 - **Splashscreen:** Permite utilizar telas de splash no aplicativo.
 - **Device:** Provê informações sobre o aparelho e sistema.
 - **File e File Transfer:** Permite trabalhar com arquivos e enviá-los para servidores remotos.
 - **E mais um monte: GPS, Browser, Bluetooth, Acelerômetro, Vibração, SMS, Touch ID, Lanterna...**

Ionic CLI

- **ionic start:** Semelhante ao *cordova create*. Gera um projeto ionic novo. Ele já virá com um template básico
- **ionic plugin, ionic platform:** Idênticos ao Cordova.
- **ionic add ionic-platform-web-client:** Adiciona interface com os serviços web do Ionic.
- **ionic browser add crosswalk:** Adiciona a biblioteca Crosswalk ao Android, que faz com que ele utilize para exibir o app um browser muito superior ao nativo.
- **ionic io init:** Inicia os serviços web do Ionic.
- **ionic serve —lab:** Abre uma janela no navegador onde você pode testar o app, que atualiza cada vez que o código mudar.
- **ionic upload:** Envia o projeto para ser testado no Ionic View.
- **ionic resource:** Gera assets para o projeto.

Ferramentas Ionic

- **Ionic View:** Aplicativo que permite você testar o aplicativo direto no seu aparelho como se ele já estivesse instalado lá. Precisa criar conta no Ionic, usar o *ionic upload* e testar. Pode ser enviado para outros testarem com o comando *ionic share *email**.
- **Ionic Push:** Envia notificações Push tanto para iOS quanto para Android sem ter que lidar com as complicações de cada um.
- **Ionic Analytics:** Gera estatísticas de uso do aplicativo. Pode ser inserido em pontos específicos do código para gerar estatísticas bem localizadas.
- **Ionic Package:** Compila o aplicativo para iOS e Android na nuvem do Ionic, sem precisar ter os SDKs instalados localmente.
- **Ionic Deploy:** Permite atualizar o aplicativo diretamente nos aparelhos dos usuários sem que precisem acessar a App Store.
- **Ionic Creator:** Permite criar protótipos de telas usando drag-and-drop para visualizar o resultado. Bastante limitado por enquanto.
- **Todos estes serviços (com exceção do View) ainda estão em Alpha.**

Assets

- Aplicativos precisam de dois assets: **Logo e Splashscreen.**
- **Cada plataforma precisa dos assets em tamanhos e formatos diferentes.**
- **Possível solução: ionic resource.**
- Logo deve ser **192x192px** e splashscreen, um quadrado **2208x2208px**.
- Se o ionic resource não funcionar, necessário mexer nas pastas de cada plataforma para colocar os assets nos locais corretos e configurá-los:
 - O Android é configurado pelo config.xml;
 - O iOS é configurado pelo programa XCode.

Compilando e Assinando

- **O processo de compilar e assinar um projeto iOS é bem simples.**
 - Comando *cordova build ios* atualiza a pasta da plataforma;
 - Lá dentro é possível abrir um projeto XCode;
 - Pelo XCode, configura-se versão atual, ícones e orientações.
 - O próprio XCode vai solicitar que seja feito o login em uma conta de desenvolvedor Apple;
 - Ele vai então baixar e gerar os devidos certificados e “provisioning profiles”.
 - Basta conectar um dispositivo iOS ao computador, rodar o comando “Archive” e está pronto.
 - Na tela “Organizer”, com mais um clique, o aplicativo é validado e enviado para a Apple.
 - No site iTunes Connect, cria-se um projeto e adiciona-se descrição, prints da tela, preço e associa-se o projeto ao app enviado.
 - Basta então aguardar uma semana para que a Apple aprove o app e ele aparecerá na loja.
- **Compilar e assinar sem XCode: Possível, porém muito mais tempo e esforço necessário.**

Compilando e Assinando

- **O processo de compilar e assinar um projeto Android também é tranquilo.**
 - Após ter certeza que todos os assets estão em ordem:
 - Rodar o comando *cordova build android —release*.
 - Isso irá gerar uma APK não assinada (sem o comando release, ela é assinada com uma chave de debug).
 - Para assiná-la, necessário ter um arquivo *keystore*.
 - Para criar um: *keytool -genkey -v -keystore nome.keystore -alias nome_alias -keyalg RSA -keysize 2048 -validity 10000*
 - Será necessário fornecer um password para o keystore e um para a chave dentro dele.
 - Para assinar o app com este keystore: *jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore nome.keystore android.apk nome_alias*
 - Agora, realinhar o APK: *zipalign -v 4 android.apk final.apk*
 - Finalmente, basta entrar no site do Android, criar uma página na loja e fazer upload do APK.

Segurança e Preço

- **Para ter um certificado de desenvolvedor Android, é necessário pagar 25\$ e ter uma licença pra sempre.**
- **Já na Apple, são 99\$ todo ano, com direito a desenvolver para Macs, iOS, Apple Watch e Apple TV.**
- Na Apple, o certificado utilizado para assinar o aplicativo está atrelado à sua conta. Ele expira a cada alguns meses por questão de segurança e também pode ser trocado se for violado.
- No Android, o certificado não está diretamente relacionado à conta. Se você perder o keystore, não poderá mais enviar atualizações do aplicativo (apenas criar um novo igual). A chave não pode ser trocada nem se for comprometida. Por esse motivo, a data de expiração é ajustada para algum valor suficientemente alto (como 2035).

Obrigado!