1. Apache Cordova

Apache Cordova é um framework de desenvolvimento móvel de código aberto que permite a utilização de tecnologias web padrão - HTML5, CSS3 e JavaScript - para o desenvolvimento multiplataforma, evitando a necessidade de uso das linguagens nativas de cada plataforma (Apache Cordova, 2015). Isto é possível porque os aplicativos são executados dentro de um invólucro particular de cada plataforma.

O Cordova é a versão software livre de um projeto da Adobe Systems Inc. Chamado PhoneGap[[1]](#footnote-1). O projeto Cordova é mantido pela Apache Software Foundation e distribuído sob a licença Apache License, versão 2.0. Atualmente na versão 6.1, o Cordova possui compatibilidade com uma ampla gama de plataformas: Android, iOS, Windows Phone, BrackBerry, Ubuntu, Firefox OS, WebOS e FireOS.

Existe uma confusão entre os termos Cordova e PhoneGap. O blog do PhoneGap (Adobe, 2012) explica que o PhoneGap, que é da Adobe, foi doado para Apache. Nesta transição o projeto mudou seu nome para Cordova, porém o PhoneGap ainda existe, com código aberto e é uma distribuição do Apache Cordova. É possível fazer uma analogia da relação do Cordova com o PhoneGap pensando no Cordova como um motor que alimento o PhoneGap, similar ao modo como o WebKit alimenta o Chrome ou Safari (Paludo, 2014).

Conforme Pietro (2012), apesar de utilizar ferramentas de desenvolvimento web, o Cordova possibilita a criação de aplicativos híbridos. Ambros (2013) explica que esse tipo de aplicativo é parcialmente nativo, parcialmente *web app*. *Web Apps* acessam funcionalidades semelhantes a um aplicativo nativo, porem vários recursos são inacessíveis pelo navegador. Aplicativos Cordova híbridos, podem aproveitar todas as funcionalidades do dispositivo.

Para entender melhor, portanto, o Cordova é um conjunto de APIs que permite que o desenvolvedor acesse funções nativas do dispositivo, como câmera, agenda, etc.; através de tecnologias web, em vez de linguagens especificas de dispositivo, ou seja, sem necessidade de escrever código em linguagem nativa (Fasciana, 2014). O desenvolvimento é como o de qualquer site e por isso oferece uma maior facilidade de aprendizado. Dispensando, por exemplo, longas horas de dedicação a sistemas mais complicados, como Java, além de reduzir os custos do projeto.

Nas aplicações Cordova toda a renderização do layout é feita através de HTML ao invés de utilizar os elementos de interface visual de cada plataforma. Na prática, a aplicação nativa gerada pelo Cordova é apenas um contêiner que interpreta o conteúdo HTML, JavaScript e CSS, de forma similar ao que um navegador *web* comum faz, exceto pelo fato de não possuir a barra de endereço ou histórico de navegação, por exemplo. Dessa forma, não existe nenhuma transformação ou geração de código a partir do conteúdo *web*, que é executado de forma interpretada no dispositivo móvel.

A figura 1 ilustra o processo de empacotamento da aplicação que o Apache Cordova faz para enviá-la para o dispositivo, ao contrário do que alguns desenvolvedores pensam, o Cordova não traduz, muito menos transforma, o código fonte em um código da linguagem nativa dos dispositivos, por exemplo Java para Android e Objective-C para iOS, o framework se utiliza da classe *WebView* para que através dela a aplicação seja executada (Android, 2015; Wargo, 2015)



Figura 1 - Empacotamento da Aplicação Cordova

Na figura 2 é demonstrado como a aplicação é executada após a instalação no dispositivo. Como se pode ver, a aplicação encontra-se dentro do *WebView,* será por ele que aplicação será executada, pode-se ver um exemplo nos aplicativos do Facebook e do Twitter onde ao clicar para abrir um link externo, nas versões mais recentes do Android, em vez de abrir o navegador, ou uma lista de navegadores para selecionar um, é aberto diretamente pela aplicação (Android, 2015; Wargo, 2015).



A aplicação utiliza-se de plug-ins para acessar as APIs nativas, esses plug-ins são codificados de duas maneiras: a primeira em JavaScript, as quais o front-end acessa, normalmente através do código em HTML; e a segunda em linguagem nativa do dispositivo, essas são chamadas pelas anteriores, em JavaScript, para executar determinadas ações normalmente relacionadas a chamadas no servidor ou acesso as APIs nativas do dispositivo, como por exemplo acessar a câmera ou o GPS (Android, 2015)

Os plugins do cordova são desenvolvidos de tal forma que um mesmo pode ser utilizado em várias plataformas, dessa forma o desenvolvedor utiliza seus métodos na parte web da aplicação, através do JavaScript (CORDOVA, 2015). Atualmente o Cordova conta com mais de 750 plugins.

Para Ambros (2013) baixo custo de desenvolvimento, facilidade de manutenção e independência de plataforma são vantagens encontradas ao desenvolver um aplicativo Cordova. As linguagens utilizadas no desenvolvimento podem ter sido previamente aprendidas e geram projetos para diferentes plataformas. Aplicativos nativos exigem conhecimentos específicos de linguagens e ferramentas do dispositivo e sistema operacional.

Como descrito anteriormente o Apache Cordova fornece um conjunto de APIs para que os aplicativos possam acessar recursos de cada dispositivo. Portanto, este *framework* é recomendado para os desenvolvedores que desejam criar aplicativos para mais de uma plataforma, inclusive com distribuição do aplicativo via as respectivas lojas de aplicativos de cada plataforma. Este *framework* também é recomendado aos desenvolvedores que procuram produzir aplicativos de maneira híbrida, ou seja, utilizando uma *WebView* para acessar componentes nativos por meio de plug-ins. No corrente projeto, foi utilizada as APIs fornecidas pelo Apache Cordova, principalmente ás referentes à câmera, a rede e a biblioteca de arquivos.

1. Plataforma para desenvolvimento de aplicativos móveis em código aberto que permite aos desenvolvedores criarem aplicações utilizando HTML5, CSS3 e JavaScript sem a necessidade de depender de APIs específicas. Os aplicativos criados são compatíveis com iOS, Windows Phone e Android. [↑](#footnote-ref-1)