

APACHE CORDOVA

Eduardo de Oliveira Castro	12/0115921
Gabriel Augusto Barbosa	09/0114281
Victor Hugo Alves de Carvalho	10/0041370
Winstein Caldeira Martins	10/0042066
Yuri Moraes Mota	10/0055852

O que é o Cordova?

- Apache Cordova é uma solução/framework open-source para construção de aplicativos móveis multi-plataforma utilizando tecnologias padrões da Web como HTML, CSS e Javascript.
- Utiliza o browser padrão do aparelho para renderizar os aplicativos como se fossem nativos da plataforma.
- Como rodam de forma nativa no celular também é possível utilizar os recursos nativos de cada plataforma/aparelho (acelerômetro, gps, etc).
- Versão atual: 3.5.0

Phonegap? Cordova?

- São basicamente a mesma coisa.
- Tudo começou como Phonegap em 2008, até que em 2011 o projeto foi comprado pela Adobe.
- Por interesses comerciais e afim de garantir para a comunidade de que o projeto continuaria sendo open-source para sempre ele foi cedido para a Apache Foundation.
- Como o nome Phonegap é uma marca registrada pela Adobe então o nome escolhido para o novo projeto foi Cordova. Hoje são dois projetos diferentes mas o Phonegap utiliza o Cordova como sua base.

Phonegap? Cordova?

- Dessa forma a Adobe conseguiu fazer com que a comunidade não se desmanchasse e continuasse contribuindo para o crescimento da plataforma. O Phonegap possui todos os recursos do Cordova mas o inverso não é válido.
- Como a Adobe possui interesses comerciais em cima da plataforma então alguns recursos, criados por eles, podem ser exclusivos da plataforma Phonegap como por exemplo o Phonegap Build, que permite a construção e execução de um aplicativo na nuvem.
- O Phonegap também é uma plataforma open-source.

Phonegap? Cordova?

- Hoje, para o desenvolvedor, não existem diferenças na hora de codificar entre as duas plataformas.
- Dica: a comunidade ainda costuma se referir ao framework como Phonegap, portanto é muito mais fácil encontrar tutoriais, plugins ou suporte da comunidade pesquisando por Phonegap e não por Cordova.

Vantagens e desvantagens

- Aplicação nativa(=ObjC, Java, etc): acesso completo a dispositivo, maior performance e marketplaces. Desenvolvimento é mais caro e não é multiplataforma.
- Aplicação web: Desenvolvimento mais barato e multiplataforma. Falta de acesso aos recursos nativos e marketplaces.
- Aplicação híbrida(=Cordova): Um mix das suas opções acima. O desenvolvedor tem acesso aos recursos de cada plataforma e suas marketplaces e o desenvolvimento é mais barato e rápido.
- A gartner acredita que até 2016 cerca de 50% das aplicações móveis vão ser híbridas(pesquisa publicada em 2013).

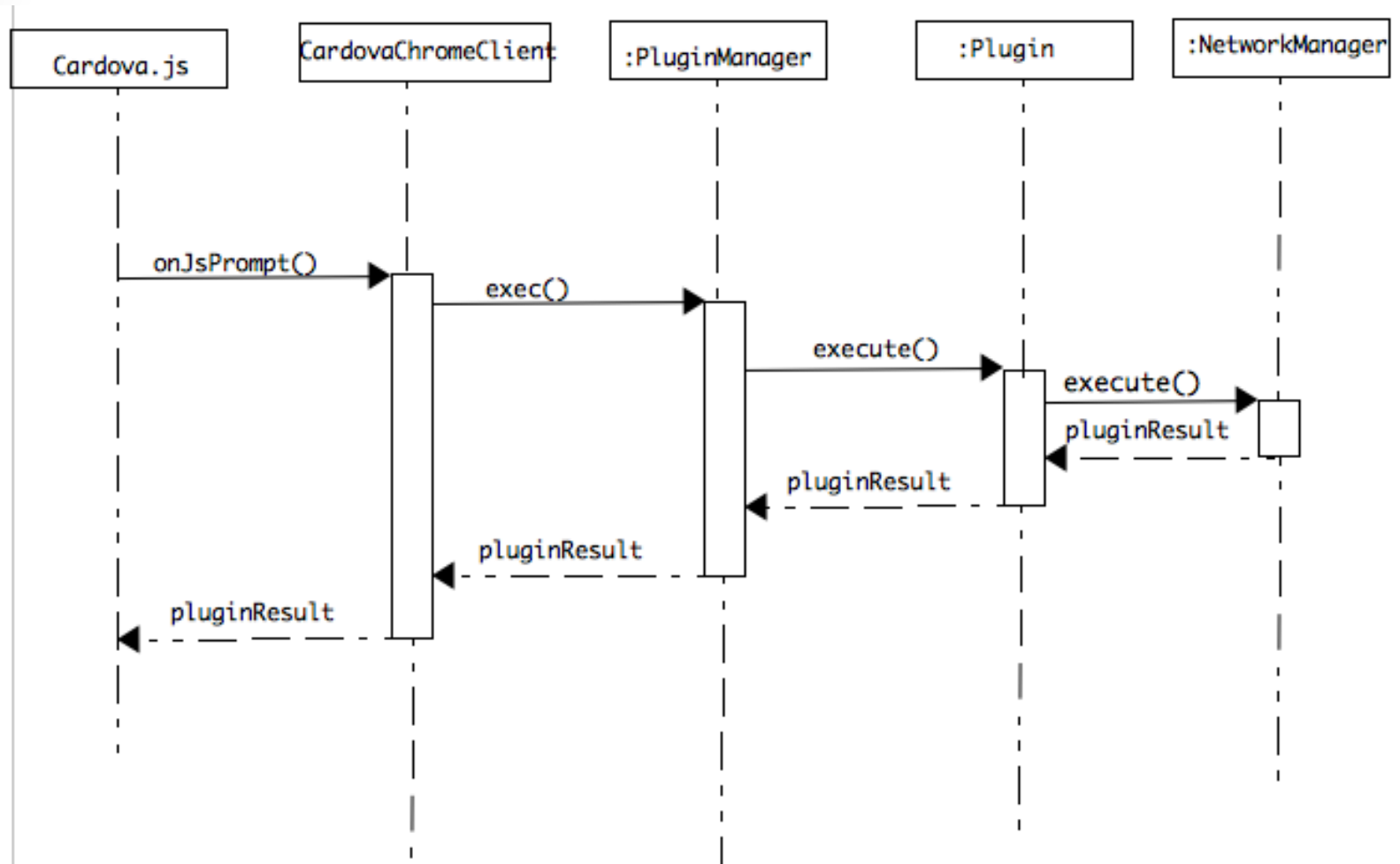
Qual o tipo de Framework?

- **Caixa Branca**
- Reuso provido por herança
 - Entender os detalhes de como o Framework funciona
 - Sobrescrita e sobrecarga de métodos
- **Caixa Preta**
 - Reuso é feito por composição
 - Não precisa entender a estrutura do Framework, precisa apenas entender a interface.
 - Combina os componentes para criar sua aplicação
- **O Cordova é um tipo de Framework caixa branca**

Pontos de Variação

- Hot-Spots
 - CordovaPlugin.class
 - CordovaActivity.class
 - Activity.class
- Frozen-Spots
 - Notification.class
 - CordovaChromeClient.class
 - InAppBrowser.class
 - InAppBrowserDialog.class
 - InAppChromeClient.class

Como funciona o Cordova?



Como funciona o Cordova?

- **Passos comunicação síncrona**

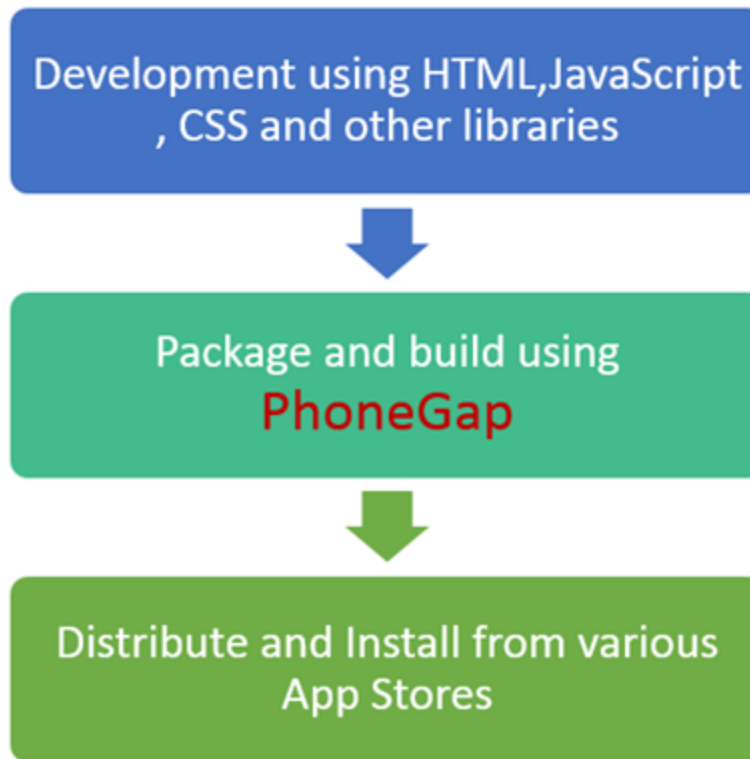
1. Cordova.exec é chamado. Esta função chama o método prompt de comando
2. Método CordovaChromeClient.onJSPrompt é interceptado
3. PluginManager.exec é chamado
4. Plugin.execute é chamado no mesmo segmento .
5. NetworkManager é chamado, o que chama a biblioteca nativa sockMan.getActiveNetworkInfo ()) e retorna PluginResult
6. CordovaChromeClient responde ao cliente via método JsPromptResult.onConfirm

Como funciona o Cordova?

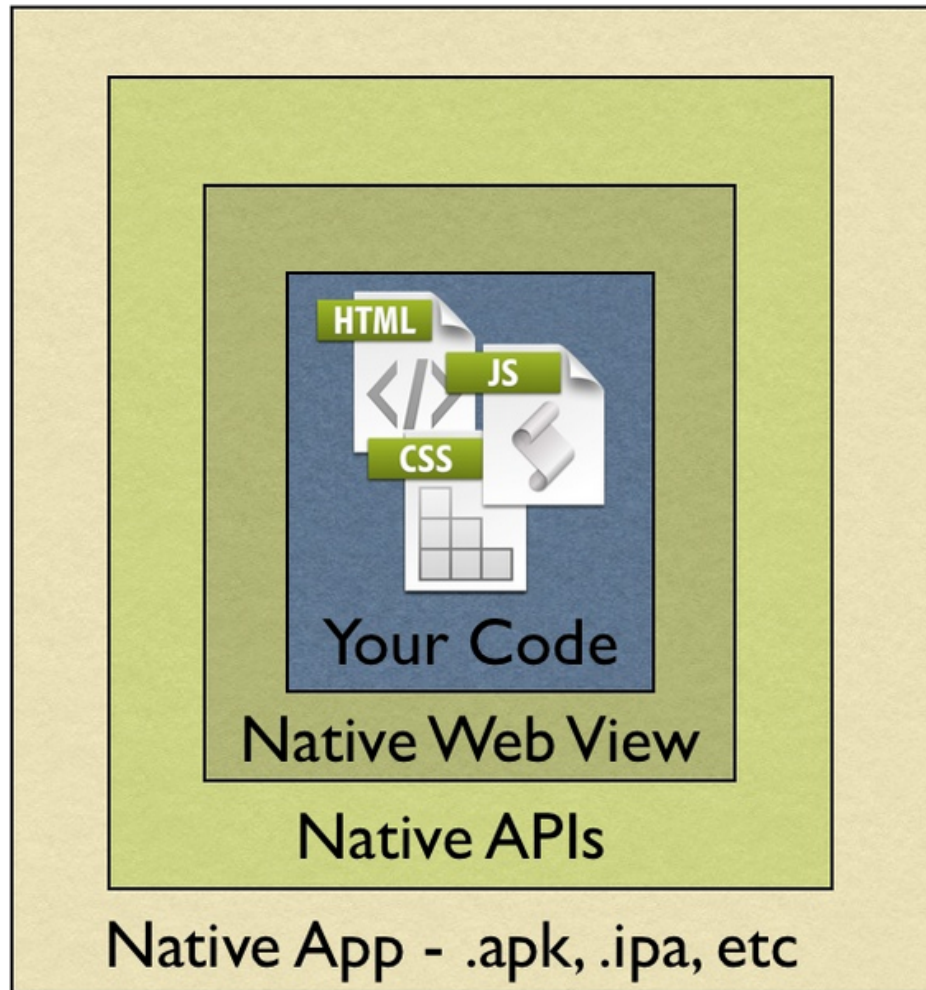
- **Passos comunicação assíncrona**

1. Seqüência acima é aplicável apenas quando o fluxo de trabalho é síncrono
2. Em caso de Plugins fluxo de trabalho é assíncrona (a menos que você o substitua)
3. Em assíncrono abaixo algumas mudanças
 - NativetoJSMODE é inicializado - consulte a seção 2 acima. Este será HANGING_GET (se carregamento url do arquivo). Ele gera XMLHttpRequest
 - Os passos 1,2 e 3 são as mesmas etapas para a comunicação síncrona)
 - Plugin.execute chamado new Thread
 - PluginManager retorna null.
 - CordovaChromeClient responde com mensagem em branco-via JSPromptResult.onconfirm
 - A resposta ao pedido do cliente (depois de chamar API nativa relevante) é propagado via CallbackServer
 - Response é analisado através da função XMLHttpRequest.onreadystatechange () e repassados para o cliente

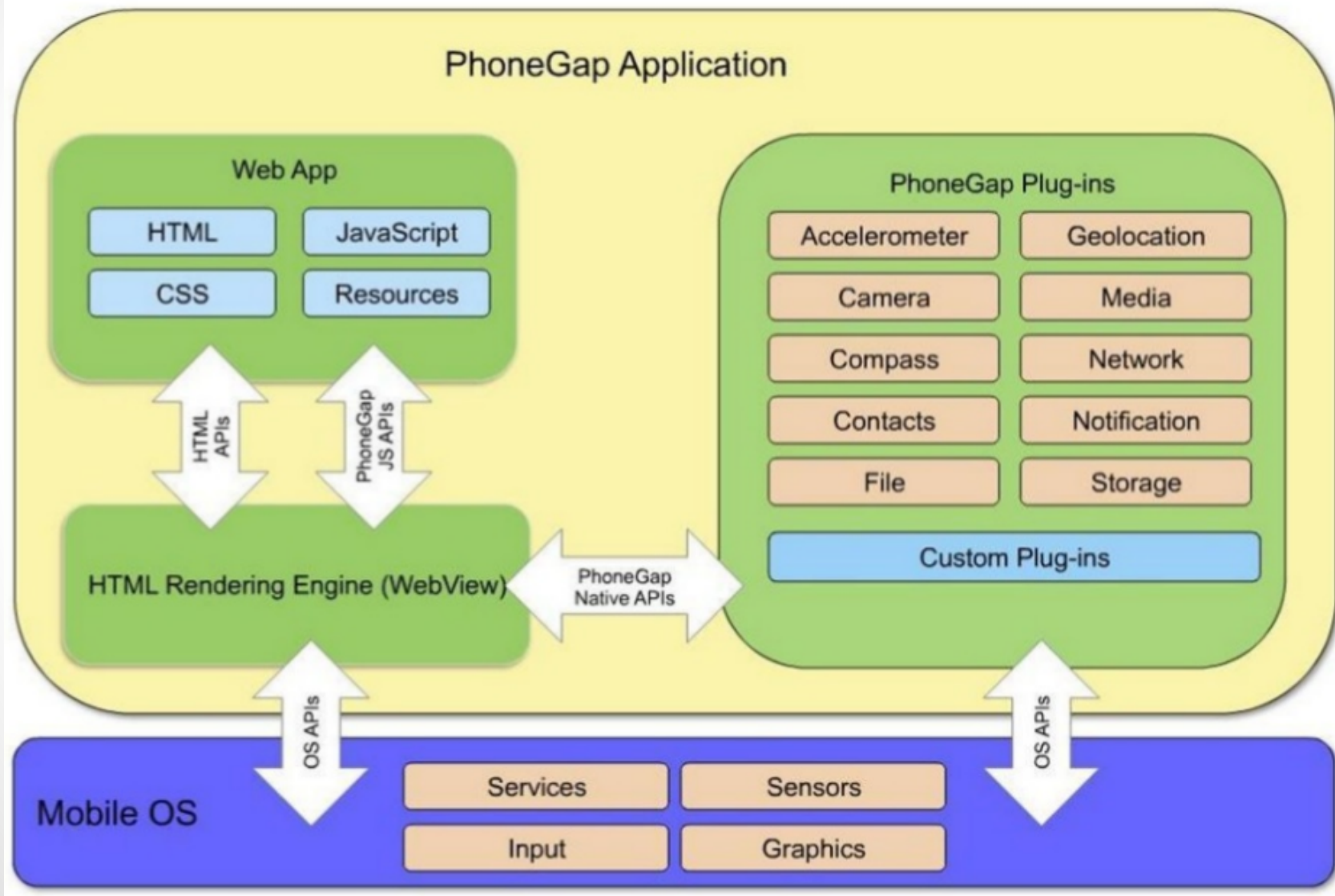
Ou simplificando...



Ou simplificando...



Arquitetura do Cordova/Phonegap



Como criar um aplicativo no Cordova?

- Para criar um projeto

```
$ cordova create hello com.example.hello HelloWorld
```

- Para adicionar suporte a uma determinada plataforma

```
$ cordova platform add ios  
$ cordova platform add amazon-fireos  
$ cordova platform add android  
$ cordova platform add blackberry10  
$ cordova platform add firefoxos
```

Como criar um aplicativo no Cordova?

- Para remover uma determinada plataforma

```
$ cordova platform remove blackberry10  
$ cordova platform rm amazon-fireos  
$ cordova platform rm android
```

- Para gerar a build de um projeto

```
$ cordova build
```

- Para gerar uma build especifica

```
$ cordova build ios
```


Como criar um aplicativo no Cordova?

- Para emular um projeto

```
$ cordova emulate android
```

- Para rodar um projeto direto no dispositivo

```
$ cordova run android
```

- Naturalmente para emular um projeto é necessário ter a SDK do mesmo instalada, logo um projeto para iOS só pode ser testado em um Mac OS X. Outra forma de testar é abrir o próprio arquivo html no browser do pc.

Como criar um aplicativo no Cordova?

- Para adicionar um plugin "oficial"

```
$ cordova plugin add org.apache.cordova.device
```

- Para adicionar um plugin "não-oficial"

```
$ cordova plugin add https://github.com/apache/cordova-plugin-console.git#r0.2.0
```

- Há plugins para cada um dos recursos nativos dos aparelhos e para muitos outros recursos como YouTube, integração com redes sociais, etc. Também é possível distribuir plugins.

Como criar um aplicativo no Cordova?

- Como estamos trabalhando com HTML, CSS e Javascript naturalmente todos os padrões já conhecidos para quem desenvolve para web também podem ser aplicados em aplicativos desenvolvidos com o Cordova.
 - Isso se estende ao uso de guias de estilo CSS, frameworks como jQuery, Bootstrap, scripts em Javascript, etc. Aqui eles são utilizados exatamente da mesma forma como seriam no desenvolvimento de uma



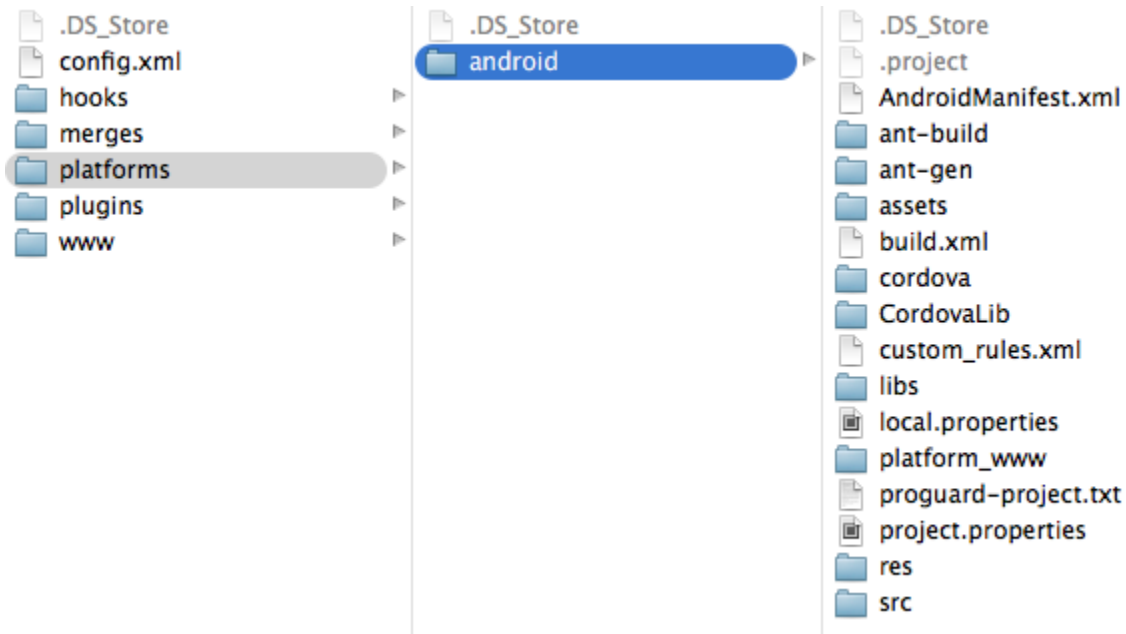
Estrutura de pastas



- config.xml: Arquivo para configurações globais do aplicativo
- /hooks: Pasta para colocarmos scripts que podem customizar o comportamento dos comandos CLI do Cordova. After build, after emulate, before run, etc.
- /merges: Pasta para arquivos de configurações específicos para cada plataforma, como por exemplo um arquivos de .css específico para o Android. Tudo o que estiver aqui vai sobrescrever os arquivos "genéricos" da pasta /www.
 - Cada plataforma deve ter uma pasta específica dentro de /merge.

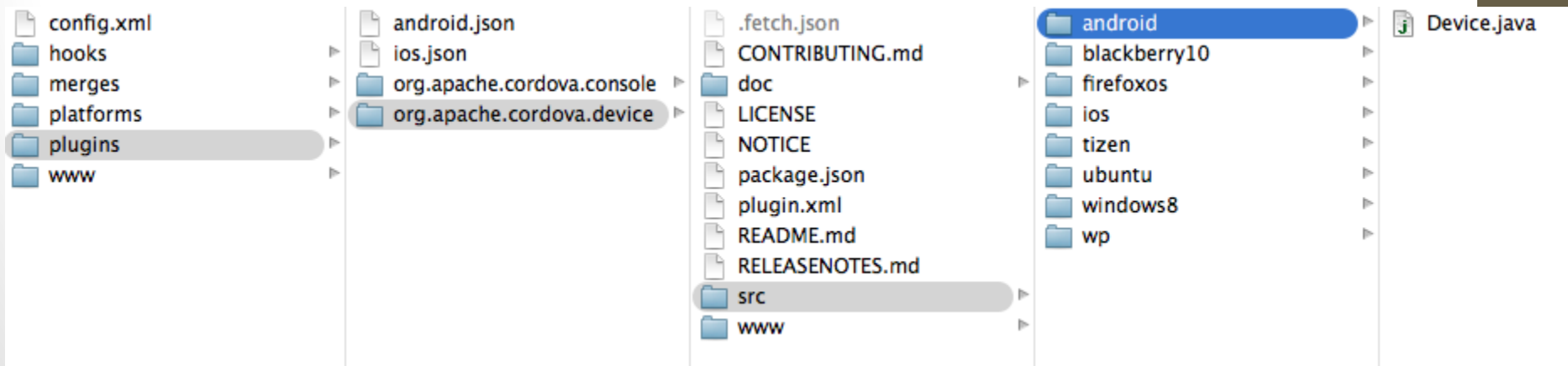
Estrutura de pastas

- Onde fica o projeto nativo para cada plataforma. A estrutura de pastas dentro de cada uma das subpastas de /platforms segue os padrões de cada plataforma



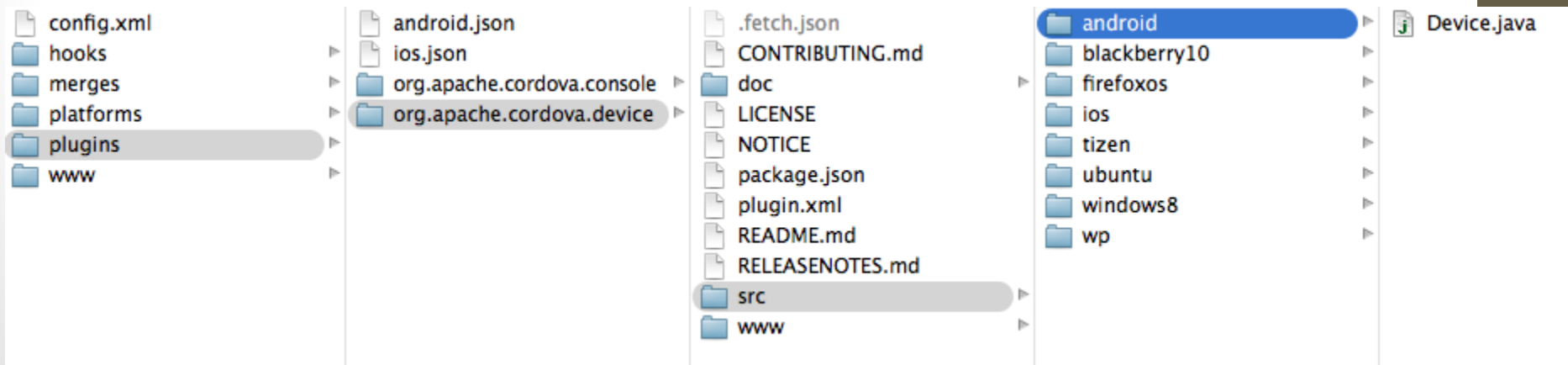
Estrutura de pastas

- /plugins: Aqui é onde colocamos todos os plugins que vão ser utilizados nos projetos. Na atual versão do Cordova não é mais necessário que nos preocupemos com essa pasta, todos os arquivos são gerenciados automaticamente pelo framework.



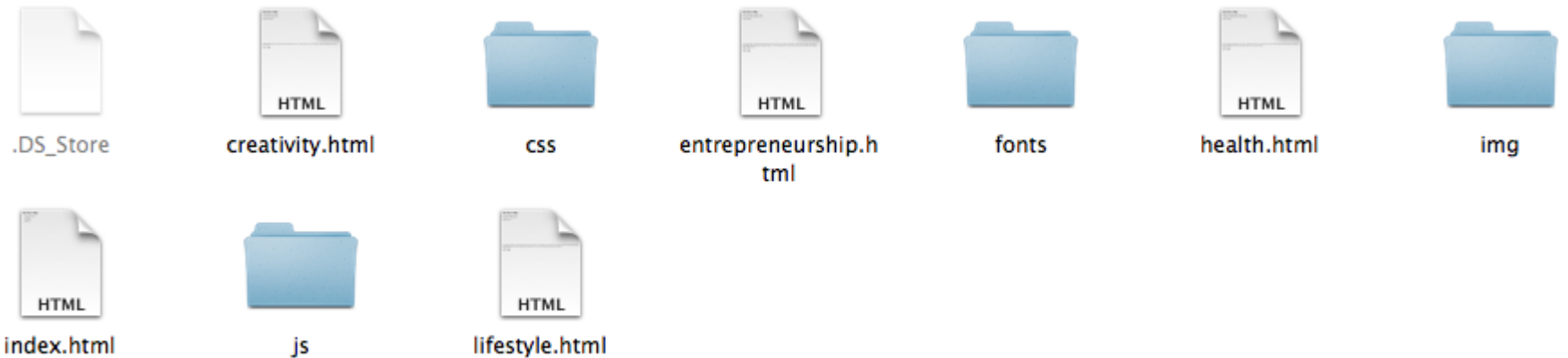
Estrutura de pastas

- Cada plugin é adicionado como uma pasta, dentro dessa pasta em src encontramos a documentação, o código nativo do plugin para cada plataforma (no caso do Android um arquivo .java) e em www encontramos um arquivo javascript de cada plugin.



Estrutura de pastas

- /www: Aqui é onde a mágica acontece, em /www que colocamos todos arquivos HTML, CSS e Javascript que vão ser trabalhados para o projeto. Quando executamos o comando 'cordova build' o que o Cordova faz é pegar o que tem nessa pasta e criar uma aplicação nativa baseada no que criamos aqui.



Estrutura de alguns arquivos

- /www/index.html

```
<head>
  <meta charset="UTF-8" />
  <meta name="format-detection" content="telephone=no" />
  <meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width, height=device-height, target-densitydpi=device-dpi" />
  <meta name="msapplication-tap-highlight" content="no" />

  <link rel="stylesheet" type="text/css" href="css/ratchet.css" />

  <script type="text/javascript" src="cordova.js"></script>
  <script type="text/javascript" src="js/ratchet.js"></script>
  <script type="text/javascript" src="js/app.js"></script>
</head>
```

- Como em uma página HTML qualquer em <head> definimos uma série de informações como scripts .js que serão utilizados, de onde retirar o guia de estilo/arquivo .css, algumas configurações para o aparelho, etc.

Estrutura de alguns arquivos

- /www/index.html

```
<script type="text/javascript" charset="utf-8">

    function onLoad()
    {
        document.addEventListener("deviceready", onDeviceReady, true);
    }

    function onDeviceReady()
    {
        // navigator.notification.alert("PhoneGap is working");
    }

    function callEntrepreneurshipPage()
    {
        window.location = "entrepreneurship.html";
    }

    function callCreativityPage()
    {
        window.location = "creativity.html";
    }

    function callHealthPage()
    {
        window.location = "health.html";
    }

    function callLifestylePage()
    {
        window.location = "lifestyle.html";
    }

</script>
```

Estrutura de alguns arquivos

- /www/index.html

```
<body onload="onLoad();">

  <div class="headerFromTheApp">
    <header class="bar bar-nav">
      <h1 class="title">One Ted A Day</h1>
    </header>
  </div>

  <div class="menuFromTheApp">
    <ul class="table-view">
      <li class="table-view-cell">
        <a class="navigate-right" onclick="callEntrepreneurshipPage()">
          <span class="badge">1</span>
          Entrepreneurship
        </a>
      </li>

      <li class="table-view-cell">
        <a class="navigate-right" onclick="callLifestylePage()">
          <span class="badge">1</span>
          Lifestyle
        </a>
      </li>

      <li class="table-view-cell">
        <a class="navigate-right" onclick="callCreativityPage()">
          <span class="badge">1</span>
          Education
        </a>
      </li>

      <li class="table-view-cell">
        <a class="navigate-right" onclick="callHealthPage()">
          <span class="badge">1</span>
          Health
        </a>
      </li>
    </ul>
  </div>

</body>
```

Estrutura de alguns arquivos

- O campo `<body>` segue os padrões de uma página web qualquer. Nesse caso estamos utilizando do framework ratchet, o "Twitter Bootstrap mobile", para o guia de estilos e fazendo chamadas de funções javascript para alternarmos as páginas.

Estrutura de alguns arquivos

- Exemplo de um projeto que utiliza do plugin VideoPlayer

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta name="viewport" content="width=320; user-scalable=yes" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8">
    <title>PhoneGap</title>
    <script type="text/javascript" charset="utf-8" src="phonegap-1.1.0.js"></script>
    <script type="text/javascript" charset="utf-8" src="video.js"></script>
  </head>
  <script type="text/javascript">

    function init(){
      document.addEventListener("deviceready", console.log('ready'), true);
    }

    function playVideo(vidUrl) {
      window.plugins.videoPlayer.play(vidUrl);
    }
  </script>

  <body onload="init();">
    <a href="#" onclick="playVideo('http://path.to.my/file.mp4')">Play HTTP</a><p/>
    <a href="#" onclick="playVideo('file:///path/to/my/file.mp4')">Play File</a><p/>
  </body>
</html>
```

Estrutura de alguns arquivos

- /platforms/android/src/com/example/onetedaday/OneTedADay.java

```
/*
    Licensed to the Apache Software Foundation (ASF) under one
    or more contributor license agreements. See the NOTICE file
    distributed with this work for additional information
    regarding copyright ownership. The ASF licenses this file
    to you under the Apache License, Version 2.0 (the
    "License"); you may not use this file except in compliance
    with the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

    Unless required by applicable law or agreed to in writing,
    software distributed under the License is distributed on an
    "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
    KIND, either express or implied. See the License for the
    specific language governing permissions and limitations
    under the License.
*/

package com.example.onetedaday;

import android.os.Bundle;
import org.apache.cordova.*;

public class OneTedADay extends CordovaActivity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        super.init();
        // Set by <content src="index.html" /> in config.xml
        super.loadUrl(Config.getStartUrl());
        //super.loadUrl("file:///android_asset/www/index.html");
    }
}
```

Estrutura de alguns arquivos

- Como o Cordova trabalha com WebViews para esse projeto foi gerado apenas um arquivo java, o Main, e nele a única coisa que o Cordova faz é chamar uma URL definida no arquivo de configurações. No caso do aplicativo usar recursos nativos eles também são chamados aqui.

Link para os repositórios dos aplicativos criados

- TED Aleatório
- One Ted A Day

<https://github.com/educastro/OneTedADayInCordova>

OBRIGADO!