



# Tecnologie e applicazioni web

## Mobile

---

Filippo Bergamasco ( [filippo.bergamasco@unive.it](mailto:filippo.bergamasco@unive.it))

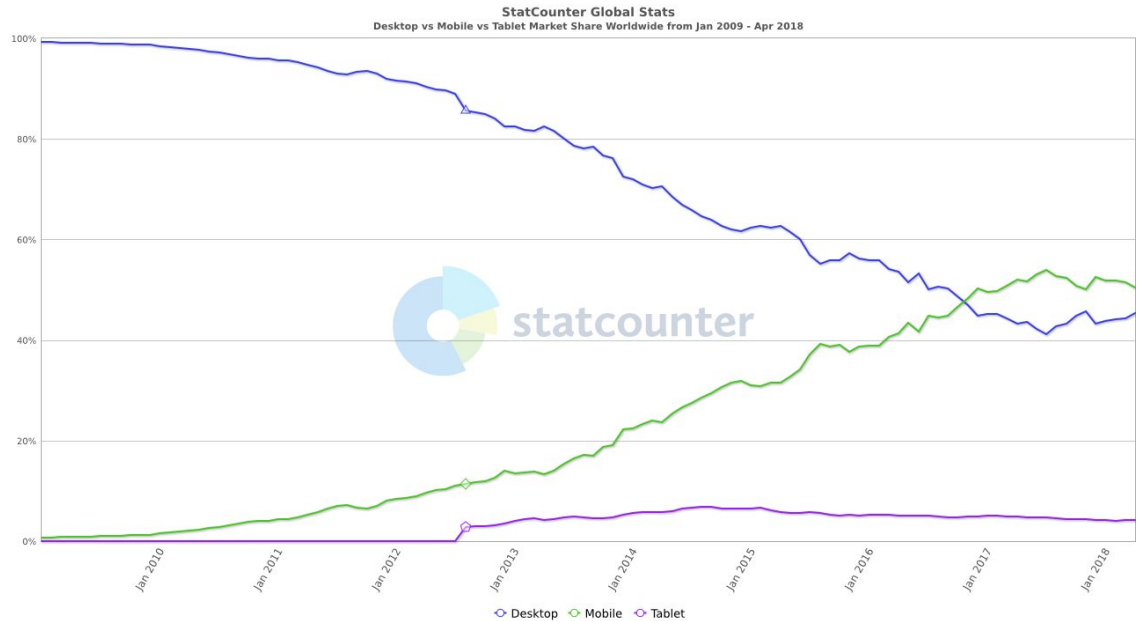
<http://www.dais.unive.it/~bergamasco/>

DAIS - Università Ca' Foscari di Venezia

Anno accademico: 2017/2018

# Mobile web

Visto l'impressionante diffusione dei dispositivi mobile, è oggi di fondamentale importanza la fetta di mercato rappresentata dagli utenti che accedono a contenuti mediante smartphone



# Mobile web

Spesso si sente parlare di “mobile web” ma in effetti non vi è alcuna differenza con il web “tradizionale”

Ciascuno smartphone oggi sul mercato dispone di un browser sufficientemente sofisticato da visualizzare qualsiasi sito web desktop

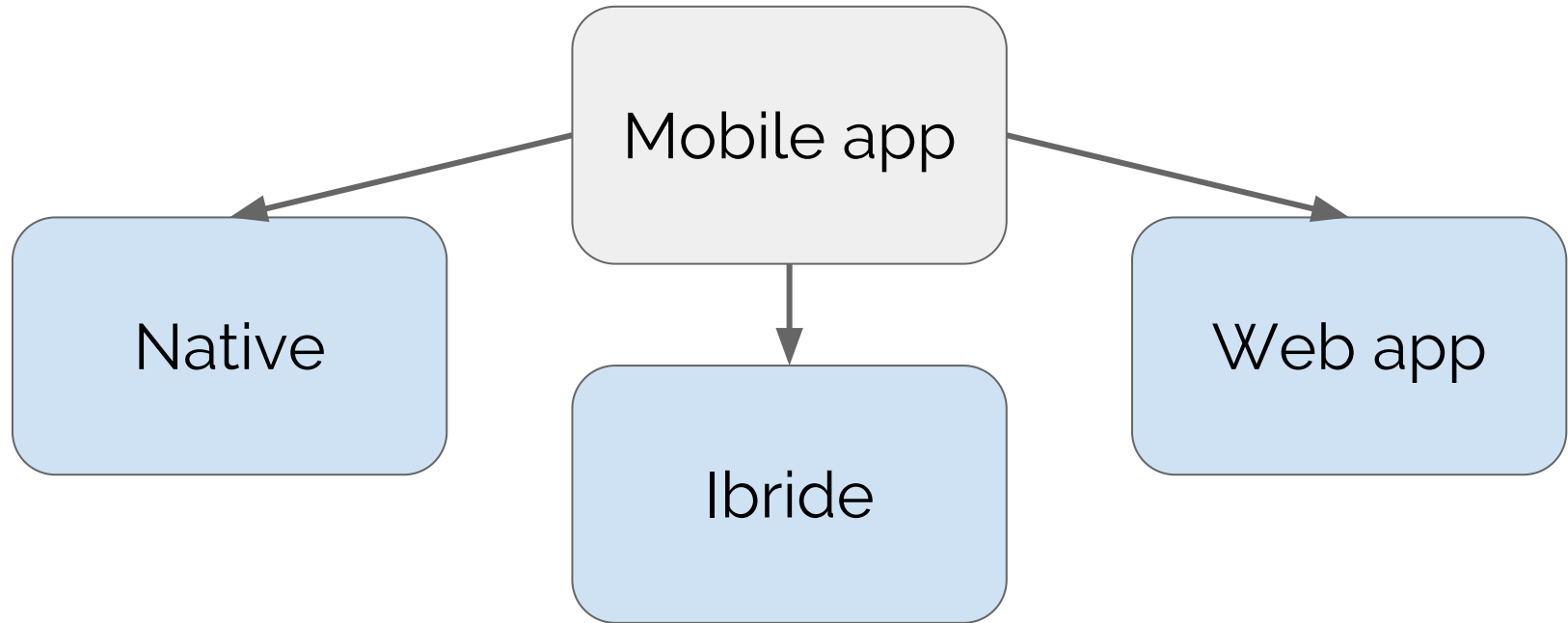
# Mobile web

La differenza principale di un app mobile web è nell'interazione con l'utente:

- Interfaccia touch based spesso senza tastiera fisica
- Display con fattori forma molto diversi dal desktop
- Risoluzione/dimensioni fisiche ridotte
- Paradigmi di interazione tipici del mondo mobile (bottoni di navigazione, toolbars, scrolling, etc)

**Problema:** grande eterogeneità nei dispositivi mobili

# Sviluppo app: cosa scegliere?



# App native

Un APP nativa è scritta utilizzando l'SDK ufficiale offerto dalla piattaforma nel linguaggio “ufficiale” in cui sono fornite le API.

Ex: Java utilizzando Android SDK, Objective-C o Swift utilizzando iOS SDK

# App native

## Vantaggi:

- Veloci e mediamente più leggere
- Accesso completo a tutte le API dell'SDK (accesso all'hardware)
- Look-and-feel coerente con il resto del sistema
- Installazione attraverso i canali ufficiali (store)

# App native

## Svantaggi:

- App multi-piattaforma richiedono un porting totale, spesso costringendo a implementare nuovamente l'intero codice
- Sviluppo mediamente più complesso
- Sviluppo costoso per App il cui scopo è principalmente quello di presentare informazioni più o meno statiche



# WebApp Mobile

Una WebApp Mobile è una web application (HTML5) che viene visualizzata/esaeguita nel browser di sistema del dispositivo utente.

Ex: Safari su iOS / Chrome su Android

# WebApp mobile

## Vantaggi:

- Utilizzano le stesse tecnologie di una WebApp desktop
- Non vanno installate, basta fornire un URL all'utente
- Direttamente multi-piattaforma perché le differenze tra i dispositivi sono gestite e risolte dal browser e non dall'applicazione stessa

# WebApp mobile

## Svantaggi:

- Look-and-feel spesso diverso da quello nativo
- Non possono accedere ad API che non sono standardizzate in HTML5
- Funzionalità limitate dalle restrizioni di sicurezza del browser
- Generalmente più lente e alto consumo di memoria

# WebApp Ibride

Una WebApp Ibrida è una web application (HTML5) che viene incapsulata all'interno di un'app nativa che fornisce una WebView (browser) full-screen.

Ex: Applicazioni create con Apache Cordova o Ionic framework

# WebApp Ibride

## Vantaggi:

- L'utente ha la sensazione di usare un'APP nativa:  
L'installazione avviene attraverso lo store
- Sviluppata usando tecnologie web (HTML5)
  - Porting di webapp "desktop" molto facilitato
  - Semplice realizzazione di app multi-piattaforma
- L'App nativa che incapsula il codice Web fornisce interfacce per accedere alle API native

# WebApp Ibride

## Svantaggi:

- Non tutte le API native sono esportate dal container
- Generalmente più lente e alto consumo di memoria (il container nativo deve fornire un browser full-featured)
- Richiede leggera customizzazione per ciascuna piattaforma implementata

# JQuery mobile

*"jQuery Mobile has appeared in the market to solve one problem: dozens of mobile platforms and browsers and the need to create compatible interfaces for all of them."*

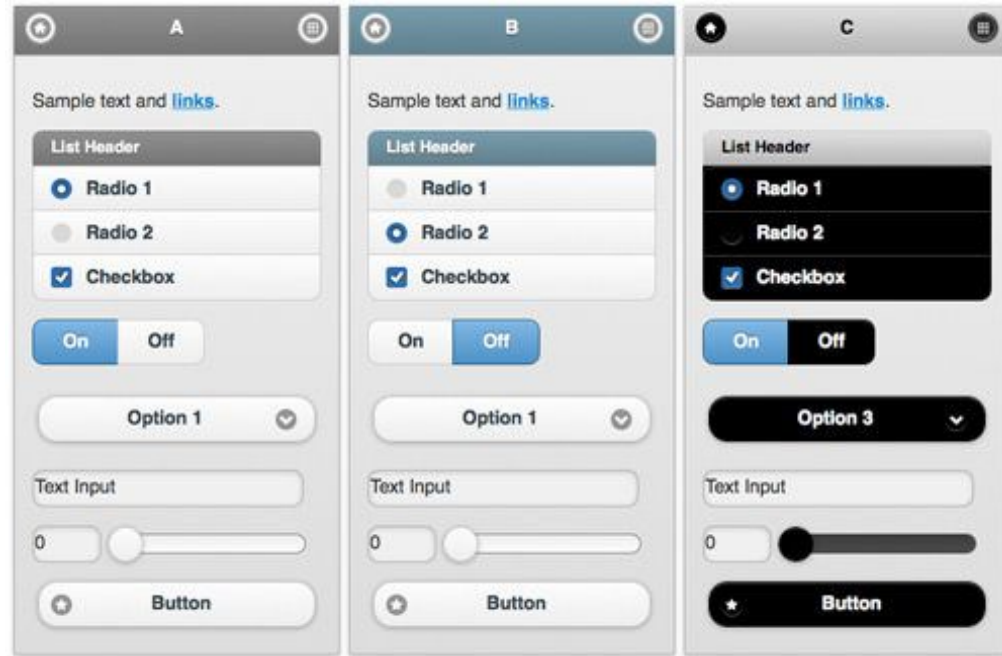
From "jQuery Mobile up and running"

## **In che senso interfacce compatibili?**

Una webapp mobile solitamente differisce da una semplice applicazione web (desktop) per il modo in cui l'interfaccia utente gestisce il workflow, emulando dove possibile l'esperienza offerta dalle applicazioni native

# JQuery mobile

JQuery Mobile fornisce un layer aggiuntivo di API HTML5 che semplifica allo sviluppatore il compito di creare interfacce con look-and-feel e paradigmi di interazione tipici del mondo mobile





# JQuery mobile non è...

## ... un alternativa a jQuery per i browsers mobile

Gestisce soltanto alcune interfacce utente (widgets) tipici del mondo mobile

## ... un SDK per creare webapp

Non permette di gestire da solo operazioni che coinvolgono features native del dispositivo

## ... un framework Javascript

La maggior parte delle funzionalità offerte possono essere implementate soltanto con codice HTML

# Utilizzo JQuery

Per utilizzare JQuery mobile è sufficiente includere i seguenti files nella struttura dell'applicazione web:

- jquery-XX.js (from the jQuery core)
- images folder
- jquery.mobile-XX\_min.js
- jquery.mobile-XX\_min.css

Se si sta sviluppando una WebApp mobile standard è possibile utilizzare i files direttamente da un CDN . Altrimenti, è necessario hostare i files sul proprio server o includerli nel container nativo nel caso di app ibride

# Template base

```
<!DOCTYPE html> <html>
<head>
<meta charset="utf-8" />
<title>Your Title</title>
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.0/jquery.mobile-1.0.min.css" />
<script src="http://code.jquery.com/jquery-1.6.4.min.js"></script>
<script type="text/javascript"
src="http://code.jquery.com/mobile/1.0/jquery.mobile-1.0.min.js">
</script>
<meta name="viewport" content="width=device-width, initial-scale=1,
user-scalable=no">
</head>
<body>
</body>
</html>
```

# Template base

```
<meta name="viewport" content="width=device-width,  
initial-scale=1, user-scalable=no">
```

La viewport è l'area visibile della WebApp. Specificando `width=device-width, initial-scale=1` impostiamo l'area visibile di modo che copra l'intera area dello schermo.

L'opzione `user-scalable=no` disabilita lo zoom del contenuto attraverso la tipica operazione pinch-zoom. In questo modo l'applicazione utilizza tutto lo spazio disponibile e si comporta in modo simile ad un app nativa

# Ruoli

jQuery mobile utilizza sintassi HTML standard per la definizione degli elementi di UI. Ciascun widget è di fatto un elemento HTML di tipo `<div>`

Per definire il comportamento e la funzionalità associata a ciascun `<div>`, jQuery mobile utilizza il concetto di ruoli, che vengono specificati attraverso l'attributo `data-role`

**Es:**

```
<div data-role="page"> </div>
```

# Ruoli

Role	Description
page	Defines a <i>page</i> , the unit that jQuery Mobile uses to show content
header	Header of a page
content	Content of a page
footer	Footer of a page
navbar	Defines a navigation bar, typically inside a header
button	Renders a visual button
controlgroup	Renders a component
collapsible	Collapsible panel of content inside a page
collapsible-set	Group of collapsible panels (accordion)
fieldcontain	Container for form fields
listview	Content of multiple items as a list
dialog	Dialog page
slider	Visual slider for Boolean values
nojs	Element that will be hidden on jQuery Mobile's compatible browsers

# Pagine

Una pagina è l'unità fondamentale in jQuery mobile che è utilizzata per visualizzare contenuti relativi ad un determinato contesto.

Un file HTML può contenere più pagine al suo interno, e soltanto una di esse può essere visualizzata in un dato istante.

jQuery Mobile fornisce un sistema di navigazione integrato che permette di cambiare pagina, gestire lo stack di pagine e caricare pagine in modo asincrono (AJAX) per rendere più fluida l'esperienza utente

# Pagine

```
<div data-role="page">  
  <div data-role="header">  
  </div>  
  <div data-role="content">  
  </div>  
  <div data-role="footer">  
  </div>  
</div>
```





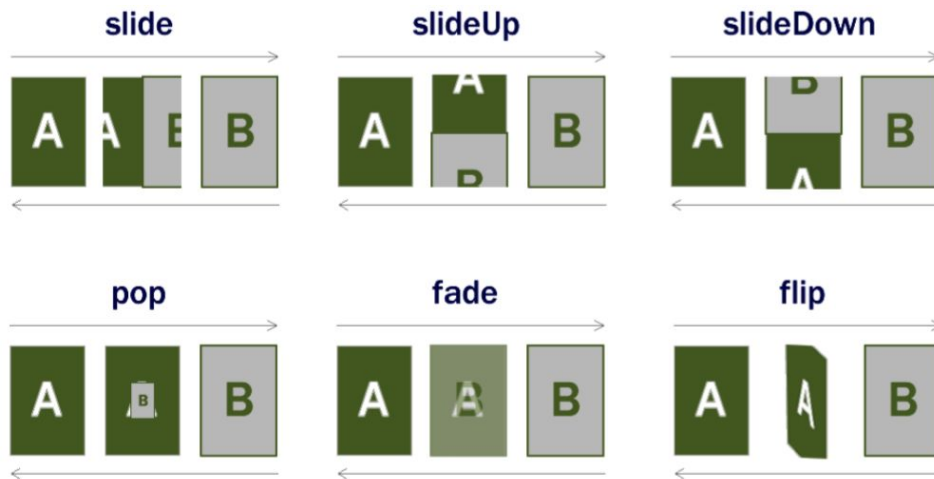
# Navigazione tra pagine

Quando si inseriscono links tra pagine diverse, jQuery Mobile ne gestisce automaticamente la transizione:

- Applicando un animazione verso la pagina successiva (e la stessa animazione in modo inverso tornando indietro)
- Aggiornando la history del browser di modo che il tasto “back” permetta di tornare alla pagina precedente
- Caricando in modo asincrono la pagina (se il DOM non è già stato caricato) e visualizzando (se necessario) un'icona di caricamento

# Transizioni tra pagine

L'attributo `data-transition="nome"` da inserire nei tag `<a>` permette di specificare il tipo di transizione da effettuare quando si cambia pagina



# Componenti di interfaccia

Per una panoramica su tutti gli elementi di interfaccia utente disponibili:

<http://demos.jquerymobile.com>



# Sviluppo app ibride



Apache Cordova è un framework open-source per sviluppare app mobile native cross-platform utilizzando tecnologie web (HTML5)

Questo tipo di applicazioni sono dette ibride perchè potenzialmente offrono il meglio fra i due mondi:

**App native:** veloci, accesso completo al device

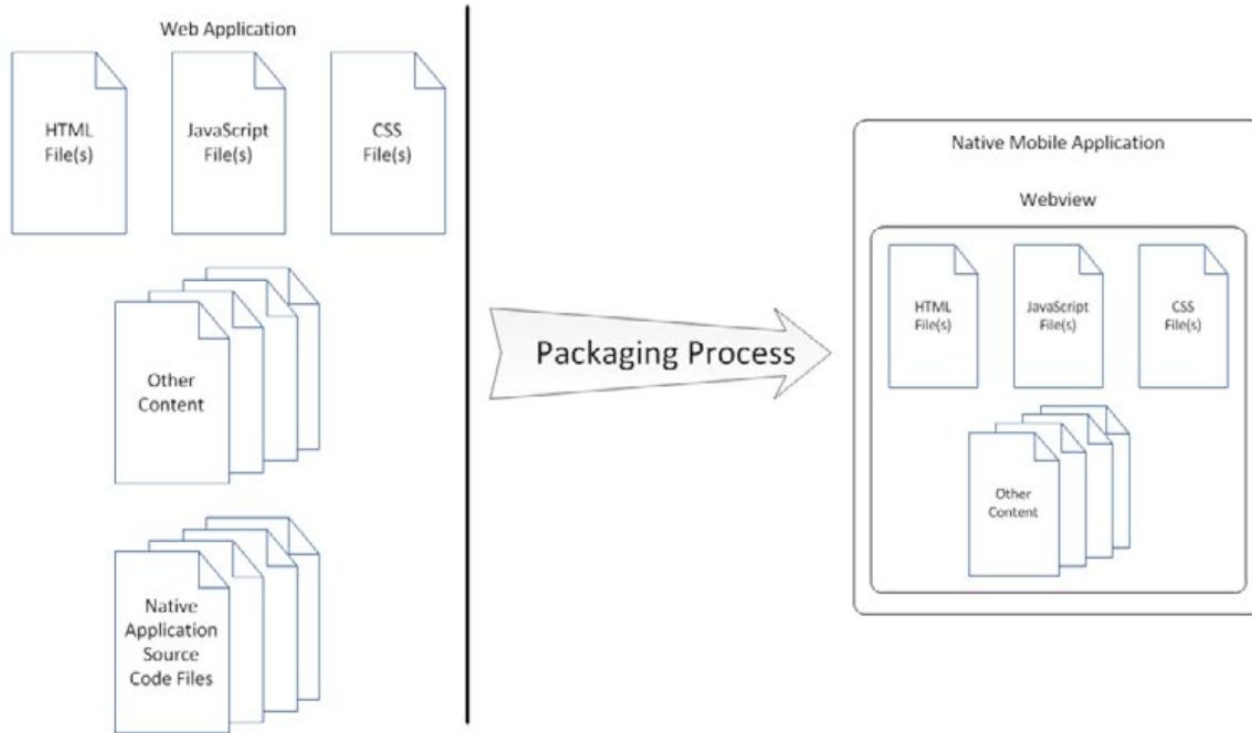
**Web App:** cross-platform, utilizzo di tecnologie note

# Apache Cordova

## Comprende

1. I sorgenti del contenitore nativo per ciascuna piattaforma supportata. Il contenitore mostra all'utente una WebView full-screen con il nostro contenuto HTML5
2. API per permettere alla nostra web-application di accedere in modo nativo al device (estendendo le funzionalità del browser)
3. Una serie di tool per gestire i plugin, il processo di build, l'esecuzione sul device fisico/emulatore, etc.

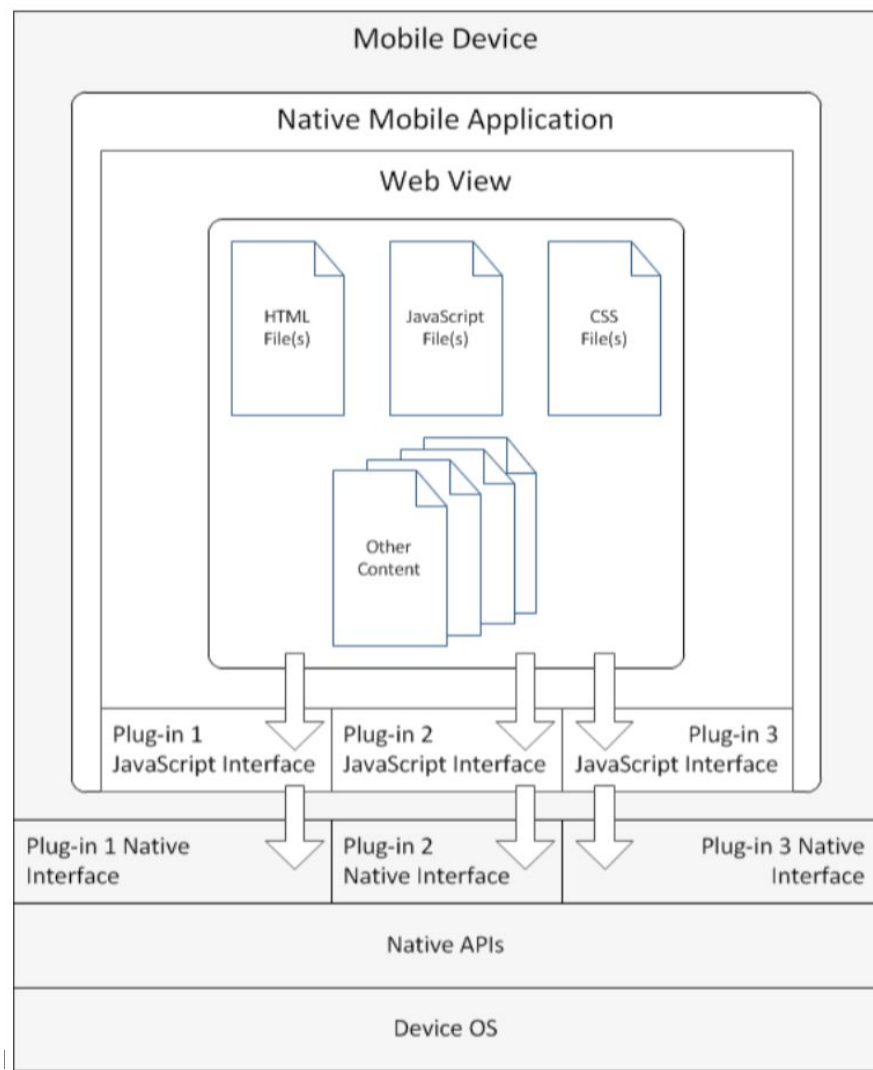
# Apache Cordova



## Cordova plugins

Il vantaggio principale di Apache Cordova rispetto ad una WebApp tradizionale è quello di permettere l'**accesso alle API native** del device **attraverso interfacce javascript (plugins)**, accessibili dalla controparte HTML 5.

In questo modo è possibile sviluppare l'intera applicazione con tecnologie web senza rinunciare alle API native



# Cordova CLI

Cordova fornisce un tool da riga di comando scritto in Javascript (per node) e distribuito attraverso npm.

Per installare:

- Installare nodejs (comprensivo di npm)
- `npm install -g cordova`

Il flag “-g” installa il pacchetto in modo globale e non nella sotto-directory `node_modules`



# Creazione nuova app

L'interfaccia CLI è chiamata semplicemente "cordova".

Per creare un nuovo progetto:

```
$ cordova create <project_path> <id>  
<nome>
```

Ad esempio:

```
$ cordova create ./myfirstapp it.unive.MyFirstApp  
myfirstapp
```

# Creazione nuova app

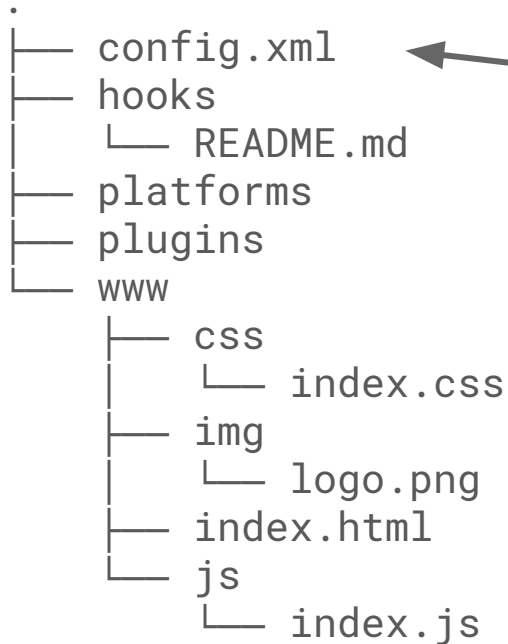
Il comando di creazione dell'app genera tutti i files necessari a realizzare la struttura base di un applicazione (HTML5)

La struttura è indipendente dalle piattaforme su cui andrà poi compilata l'applicazione (iOS, Android, etc)

Per aggiungere una nuova piattaforma:

```
$ cordova platform add <nome_piattaforma>
```

# Struttura di un progetto

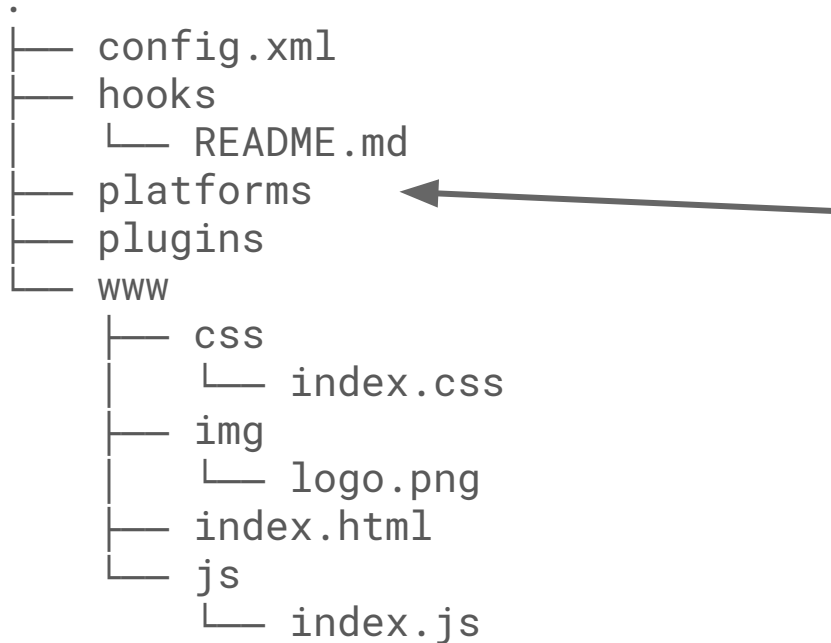


File di configurazione globale  
che controlla il comportamento  
dell'applicazione

Contiene informazioni comuni  
per tutte le piattaforme e dettagli  
specifici validi per ciascuna di  
esse

7 directories, 6 files

# Struttura di un progetto

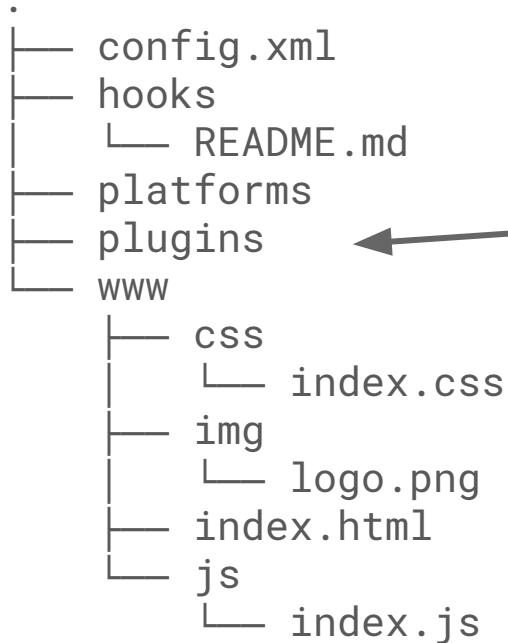


Contiene la controparte “nativa” dell'applicazione (sorgenti e files compilati) per ciascuna delle piattaforme aggiunte al progetto

Non è necessario agire manualmente sul contenuto poiché i files al suo interno vengono sovrascritti dal processo di build

7 directories, 6 files

# Struttura di un progetto



Contiene i plugins cordova che possono essere installati on-demand e aggiungono funzionalità specifiche all'applicazione.

Ex:

Plugins per l'accesso al GPS, filesystem, durata della batteria, etc.

7 directories, 6 files

# Struttura di un progetto

```
.
├── config.xml
├── hooks
│   └── README.md
├── platforms
├── plugins
└── www
    ├── css
    │   └── index.css
    ├── img
    │   └── logo.png
    ├── index.html
    ├── js
    │   └── index.js
```

Contiene i sorgenti (HTML5) della nostra applicazione. I files saranno copiati in una sotto-directory di platform in fase di build e inseriti nel pacchetto da deployare sul device

7 directories, 6 files

# config.xml

config.xml è un file di configurazione globale dell'applicazione che controlla molti aspetti dell'applicazione. La radice del file XML è un elemento di tipo “widget” che descrive:

- L'id dell'applicazione
- Il suo nome
- Qual'è il file html principale (l'index)
- Configurazioni dei plugins e specifiche per ciascuna piattaforma

[https://cordova.apache.org/docs/en/latest/config\\_ref/index.html](https://cordova.apache.org/docs/en/latest/config_ref/index.html)

# config.xml

```
<?xml version='1.0' encoding='utf-8'?>
<widget id="io.cordova.hellocordova" version="0.0.1" xmlns="http://www.w3.org/ns/widgets"
xmlns:cdv="http://cordova.apache.org/ns/1.0">
  <name>HelloCordova</name>
  <description>A sample Apache Cordova application that responds to the deviceready event.</description>
  <author email="dev@cordova.apache.org" href="http://cordova.io">
    Apache Cordova Team
  </author>
  <content src="index.html" />
  <access origin="*" />
  <allow-intent href="http://*/*" /> <allow-intent href="https://*/*" /> <allow-intent href="tel:*" />
<allow-intent href="sms:*" /><allow-intent href="mailto:*" />
  <platform name="android">
    <allow-intent href="market:*" />
  </platform>
  <platform name="ios">
    <allow-intent href="itms:*" />
    <allow-intent href="itms-apps:*" />
  </platform>
</widget>
```



# config.xml

Una delle funzionalità importanti che si possono impostare dal file config.xml è quello dell'icona dell'applicazione.

Metodo generale:

Aggiungere all'interno di widget la seguente definizione:

```
<icon src="res/icon.png" />
```

In realtà in tutte le piattaforme moderne la gestione delle icone è complicata perché deve tener conto della densità di pixel dello schermo

# config.xml: icona

```
<platform name="android">
  <!--
    ldpi      : 36x36 px
    mdpi      : 48x48 px
    hdpi      : 72x72 px
    xhdpi     : 96x96 px
    xxhdpi    : 144x144 px
    xxxhdpi   : 192x192 px
  -->
  <icon src="res/android/ldpi.png" density="ldpi"/>
  <icon src="res/android/mdpi.png" density="mdpi"/>
  <icon src="res/android/hdpi.png" density="hdpi"/>
  <icon src="res/android/xhdpi.png" density="xhdpi"/>
  <icon src="res/android/xxhdpi.png" density="xxhdpi"/>
  <icon src="res/android/xxxhdpi.png" density="xxxhdpi"/>
</platform>
```

```
<platform name="ios">
  <!-- iOS 8.0+ -->
  <!-- iPhone 6 Plus -->
  <icon src="res/ios/icon-60@3x.png" width="180" height="180" />
  <!-- iOS 7.0+ -->
  <!-- iPhone / iPod Touch -->
  <icon src="res/ios/icon-60.png" width="60" height="60" />
  <icon src="res/ios/icon-60@2x.png" width="120" height="120" />
  <!-- iPad -->
  <icon src="res/ios/icon-76.png" width="76" height="76" />
  <icon src="res/ios/icon-76@2x.png" width="152" height="152" />
  <!-- Spotlight Icon -->
  <icon src="res/ios/icon-40.png" width="40" height="40" />
  <icon src="res/ios/icon-40@2x.png" width="80" height="80" />
  <!-- iOS 6.1 -->
  <!-- iPhone / iPod Touch -->
  <icon src="res/ios/icon.png" width="57" height="57" />
  <icon src="res/ios/icon@2x.png" width="114" height="114" />
  <!-- iPad -->
  <icon src="res/ios/icon-72.png" width="72" height="72" />
  <icon src="res/ios/icon-72@2x.png" width="144" height="144" />
  <!-- iPhone Spotlight and Settings Icon -->
  <icon src="res/ios/icon-small.png" width="29" height="29" />
  <icon src="res/ios/icon-small@2x.png" width="58" height="58" />
  <!-- iPad Spotlight and Settings Icon -->
  <icon src="res/ios/icon-50.png" width="50" height="50" />
  <icon src="res/ios/icon-50@2x.png" width="100" height="100" />
  <!-- iPad Pro -->
  <icon src="res/ios/icon-83.5@2x.png" width="167"
height="167"/>
</platform>
```

# Eventi

Cordova genera molteplici eventi Javascript durante il ciclo di vita di un applicazione

Il tipo degli eventi disponibili dipende dalla piattaforma specifica

E' possibile catturare gli eventi inserendo una funzione di callback mediante l'istruzione  
`document.addEventListener( )`

# Eventi

Supported Platforms/ Events	android	blackberry10	ios	Windows Phone 8	Windows
<b>deviceready</b>	✓	✓	✓	✓	✓
<b>pause</b>	✓	✓	✓	✓	✓
<b>resume</b>	✓	✓	✓	✓	✓
<b>backbutton</b>	✓	✓	✗	✗	✓
<b>menubutton</b>	✓	✓	✗	✗	✗
<b>searchbutton</b>	✓	✗	✗	✗	✗
<b>startcallbutton</b>	✗	✓	✗	✗	✗
<b>endcallbutton</b>	✗	✓	✗	✗	✗
<b>volumedownbutton</b>	✓	✓	✗	✗	✗
<b>volumeupbutton</b>	✓	✓	✗	✗	✗
<b>activated</b>	✗	✗	✗	✗	✓

# Eventi: Deviceready

E' l'evento più importante che viene generato quando tutto l'ecosistema Cordova è stato completamente caricato. In particolare segnala che tutti i plugins sono stati caricati e sono pronti all'utilizzo

Ex:

```
document.addEventListener("deviceready", onDeviceReady, false);

function onDeviceReady() {
    // Now safe to use device APIs
}
```

# Eventi

## Pause:

generato quando la piattaforma nativa mette l'applicazione in background.

## Resume:

generato quando l'applicazione esce dal background e torna ad essere presentata all'utente

# Eventi

## Menubutton:

generato quando viene premuto il tasto “menu” sul dispositivo (se disponibile)

## Volumeupbutton, Volumedownbutton:

generati quando vengono premuti i tasti per alzare o abbassare il volume

# Building

Per compilare ed eseguire le applicazioni è necessario installare l'SDK per ciascuna piattaforma target che si desidera utilizzare

E' possibile visualizzare i requisiti richiesti da cordova con il comando

```
$ cordova requirements
```



# Piattaforma Android

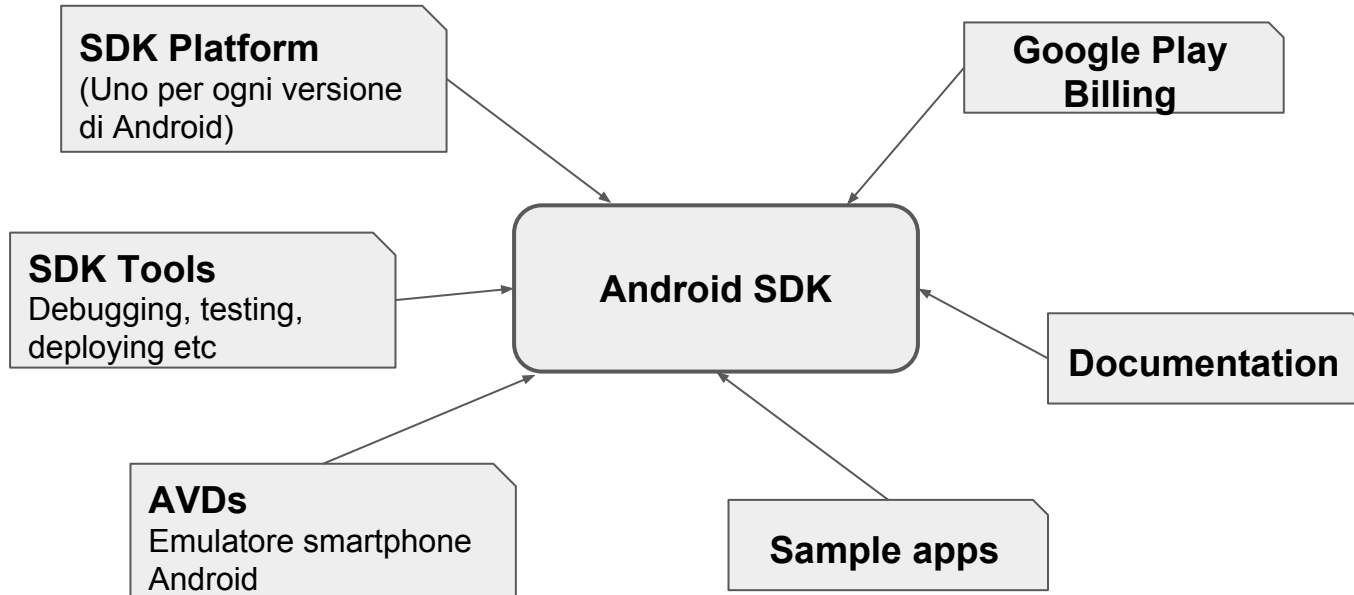
Per sviluppare applicazioni su piattaforma Android è necessario scaricare l'Android SDK fornito insieme all'IDE Android Studio

<https://developer.android.com/studio/index.html>

Il processo di build è gestito da Cordova, quindi non sarà necessario agire direttamente su Android Studio se non per gestire SDKs ed AVDs

# Android SDK

L'Android Software Developer Kit (SDK) contiene tutti i tools necessari per sviluppare apps per Android



# Android SDK Manager

Attraverso l'SDK Manager di Android Studio è possibile installare:

- Uno o più SDK Platform (librerie) per ciascuna versione di Android
- Gli SDK e Platform tools (per il processo di build e firma)
- Componenti aggiuntivi come Intel x86 Emulator Accelerator (HAXM), i driver USB di Google per il debugging, Google Play Services, etc.

E' inoltre possibile eseguire un tool chiamato AVD Manager per gestire gli AVD

# SDK Platforms

SDK Platforms

SDK Tools

SDK Update Sites

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

	Name	API Level	Revision	Status
<input type="checkbox"/>	Android P Preview	P	2	Not installed
<input checked="" type="checkbox"/>	Android 8.1 (Oreo)	27	1	Installed
<input type="checkbox"/>	Android 8.0 (Oreo)	26	2	Partially installed
<input type="checkbox"/>	Android 7.1.1 (Nougat)	25	3	Not installed
<input type="checkbox"/>	Android 7.0 (Nougat)	24	2	Not installed
<input type="checkbox"/>	Android 6.0 (Marshmallow)	23	3	Not installed
<input type="checkbox"/>	Android 5.1 (Lollipop)	22	2	Not installed
<input type="checkbox"/>	Android 5.0 (Lollipop)	21	2	Not installed
<input type="checkbox"/>	Android 4.4W (KitKat Wear)	20	2	Not installed
<input type="checkbox"/>	Android 4.4 (KitKat)	19	4	Not installed
<input type="checkbox"/>	Android 4.3 (Jelly Bean)	18	3	Not installed
<input type="checkbox"/>	Android 4.2 (Jelly Bean)	17	3	Not installed
<input type="checkbox"/>	Android 4.1 (Jelly Bean)	16	5	Not installed
<input type="checkbox"/>	Android 4.0.3 (IceCreamSandwich)	15	5	Not installed
<input type="checkbox"/>	Android 4.0 (IceCreamSandwich)	14	4	Not installed
<input type="checkbox"/>	Android 3.2 (Honeycomb)	13	1	Not installed
<input type="checkbox"/>	Android 3.1 (Honeycomb)	12	3	Not installed
<input type="checkbox"/>	Android 3.0 (Honeycomb)	11	2	Not installed
<input type="checkbox"/>	Android 2.3.3 (Gingerbread)	10	2	Not installed
<input type="checkbox"/>	Android 2.3 (Gingerbread)	9	2	Not installed
<input type="checkbox"/>	Android 2.2 (Froyo)	8	3	Not installed

# Versioni Android

Ciascuna versione è caratterizzata da:

- un nome in codice (ispirato a nomi di dolci)
- un numero di versione numerico
- L' **API level**: un identificativo numerico che specifica la versione delle API utilizzabili dalle apps.

A ciascun numero di versione corrisponde un API level. Il nome in codice spesso raggruppa più versioni (in caso di revisioni minori)

# SDK Tools

SDK Platforms <b>SDK Tools</b> SDK Update Sites			
Below are the available SDK developer tools. Once installed, Android Studio will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.			
	Name	Version	Status
<input type="checkbox"/>	Android SDK Build-Tools 28-rc1		Update Available: 28.0.0 rc1
<input type="checkbox"/>	GPU Debugging tools		Not Installed
<input type="checkbox"/>	CMake		Not Installed
<input type="checkbox"/>	LLDB		Not Installed
<input type="checkbox"/>	Android Auto API Simulators	1	Not installed
<input type="checkbox"/>	Android Auto Desktop Head Unit emulator	1.1	Not installed
<input checked="" type="checkbox"/>	Android Emulator	27.1.12	Installed
<input checked="" type="checkbox"/>	Android SDK Platform-Tools	27.0.1	Installed
<input checked="" type="checkbox"/>	Android SDK Tools	26.1.1	Installed
<input type="checkbox"/>	Documentation for Android SDK	1	Not installed
<input type="checkbox"/>	Google Play APK Expansion library	1	Not installed
<input type="checkbox"/>	Google Play Licensing Library	1	Not installed
<input type="checkbox"/>	Google Play services	48	Not installed
<input type="checkbox"/>	Google Web Driver	2	Not installed
<input type="checkbox"/>	Instant Apps Development SDK	1.1.0	Not installed
<input checked="" type="checkbox"/>	Intel x86 Emulator Accelerator (HAXM installer)	6.2.1	Installed
<input type="checkbox"/>	NDK	16.1.4479499	Not installed
▼ <input checked="" type="checkbox"/>	<b>Support Repository</b>		
<input checked="" type="checkbox"/>	ConstraintLayout for Android		Installed
<input checked="" type="checkbox"/>	Solver for ConstraintLayout		Installed
<input checked="" type="checkbox"/>	Android Support Repository	47.0.0	Installed
<input checked="" type="checkbox"/>	Google Repository	5.0	Installed

# Android Virtual Device (AVD)

E' possibile eseguire l'applicazione Android che si sta sviluppando su:

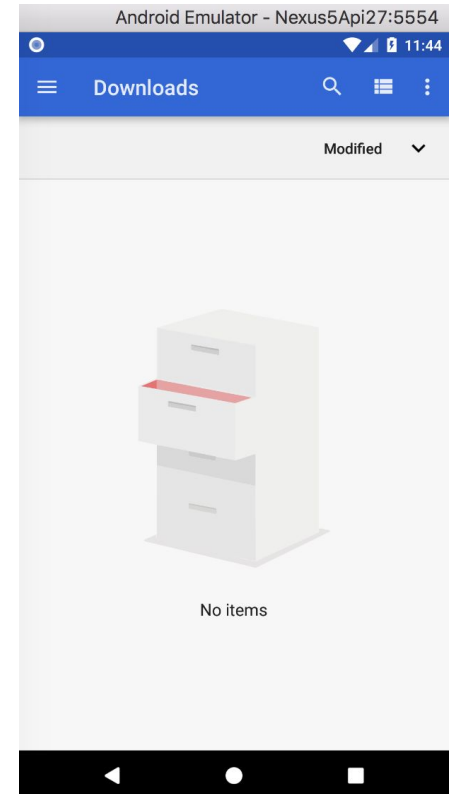
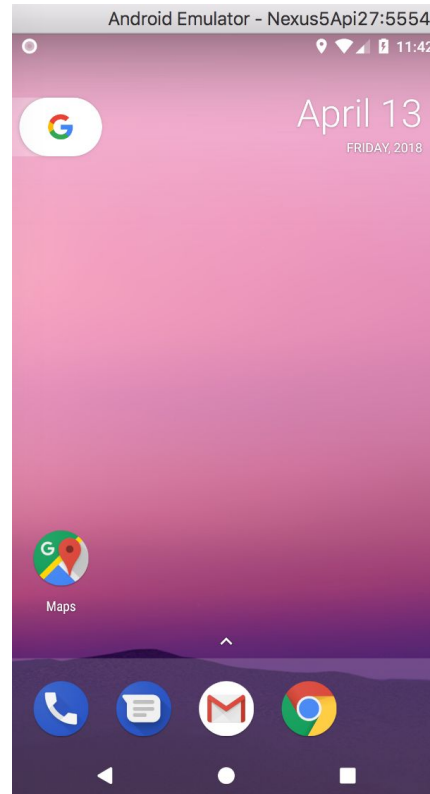
- Un device fisico, abilitando la modalità sviluppo sul device e installando i driver USB per la connessione al device
- Un device virtuale (AVD) che viene eseguito sull'emulatore Android fornito con l'SDK

Per ottenere performance accettabili, è raccomandabile installare una "system image" per CPU Intel x86 e avere l'accelerazione Intel HAXM abilitata

E' possibile creare molteplici AVD con l'AVD Manager

# Android Virtual Device (AVD)

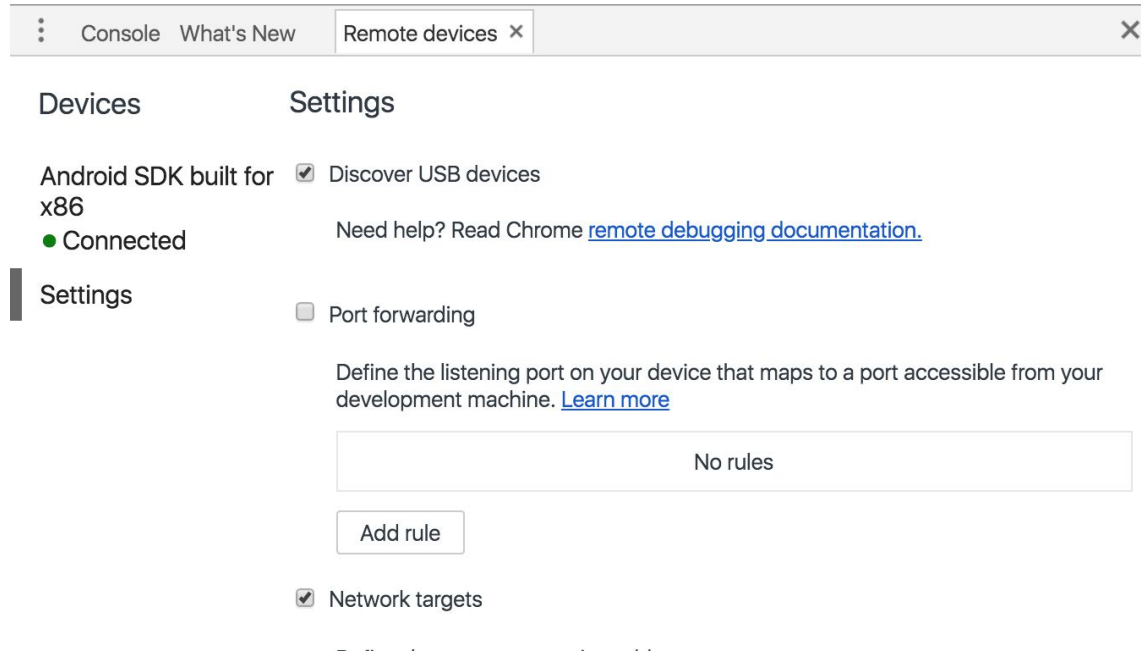
Una volta installato e configurato, un AVD può essere utilizzato come un normale dispositivo Android connesso alla porta USB del PC host





# AVD debugging

Se nel device sta girando il browser Chrome (in una WebView o di sistema) è possibile effettuare il debug remoto dai “Developer Tools” del browser Chrome host



# Android platform

Una volta installato Android SDK e installata almeno una piattaforma è sufficiente:

1. Esportare la variabile d'ambiente `ANDROID_HOME` con la directory relativa all' Android SDK
2. Aggiungere `ANDROID_HOME/platform-tools` al path corrente

Eseguire il comando:

```
$ cordova platform add android
```

La directory `platform/android` sarà automaticamente generata con tutti i sorgenti necessari a compilare l'applicazione Android

# Build

Per compilare l'applicazione e creare il pacchetto APK pronto per essere installato:

```
$ cordova build android
```

Oppure

```
$ cordova build android --release
```

Per eseguire l'applicazione (Su AVD o un device fisico)

```
$ cordova run android
```

# Plugins

La gestione delle API native avviene attraverso dei plugins che possono essere installati on-demand

Lista plugins disponibili:

<https://cordova.apache.org/plugins/>

## Plugins “ufficiali”:

Battery, camera, console, contacts, device, device motion, device orientation, dialog, file, geolocation, globalization, vibration, statusbar, etc..

# Utilizzo plugins

Per installare un plugin:

```
$ cordova plugin add <nome_plugin>
```

Alcuni plugins funzionano senza utilizzo di codice Javascript ma soltanto agendo sul config.txt

Es: cordova-plugin-statusbar

```
<preference name="StatusBarBackgroundColor"  
value="#000000" />
```

Alcuni plugins aggiungono metodi e funzioni globali accessibili via Javascript (fare riferimento alla documentazione)

# Angular to mobile

Supponiamo di avere realizzato un'applicazione web con Angular e volerla trasformare in un'applicazione mobile Ibrida

## Passaggi:

1. Modifica dei sorgenti per includere le API fornite da cordova (gestione eventi, etc)
2. Packaging dell'applicazione nella dir `www/`
3. `cordova build`

# Step 1

Caricare la libreria cordova.js all'interno dell'applicazione. Nel file index.html, inserire:

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

**Nota:** la libreria non è presente tra i sorgenti ma è resa disponibile quando l'app viene eseguita nel container nativo

# Step 2

Modificare l'indirizzo di base per il routing. Dato che l'app è servita in locale dal container, e non da un webserver, occorre modificare il tag `<base>` in questo modo:

```
<base href=". /">
```



# Step 3

Effettuare il bootstrap di Angular soltanto dopo la ricezione dell'evento `deviceready`. Modificare `main.ts` nel modo seguente:

```
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';
import { environment } from './environments/environment';

if (environment.production) {
  enableProdMode();
}
// Device bootstrap
document.addEventListener('deviceready', () => {
  platformBrowserDynamic().bootstrapModule(AppModule).catch(err =>
console.log(err));
}, false);
```

# Step 3

**Nota:** in questo modo l'app non funziona più sul browser desktop ma si garantisce che tutti i plugins cordova sono caricati prima del bootstrap di Angular

```
// Device bootstrap
document.addEventListener('deviceready', () => {
    platformBrowserDynamic().bootstrapModule(AppModule).catch(err =>
console.log(err));
}, false);
```

# Step 4

Le applicazioni Android che girano sull'emulatore o su un device fisico sono solitamente connesse alla rete attraverso un bridge creato dal sistema host.

Sull'emulatore Android, il pc host è accessibile con l'indirizzo ip 10.0.2.2

```
public url = 'http://localhost:8080';  
public url = 'http://10.0.2.2:8080';
```

# Step 5

Aprire il file `.angular-cli.json` e modificare la property `outDir` da `dist/` a `<app-root>/www/`

In questo modo, quando viene eseguito il comando  
`$ ng build`

L'applicazione Angular viene compilata impacchettata all'interno della dir `<app-root>/www`

# Step 6

Aggiungere le piattaforme mobile desiderate. Ad esempio, per la piattaforma Android:

```
$ cordova platform add android
```

Compilare l'applicazione, ed eseguirla

```
$ cordova build
```

```
$ cordova run
```

# ionic

Il front-end Angular è particolarmente indicato per lo sviluppo di applicazioni mobile

Accoppiare Angular con il container nativo fornito da Apache Cordova è così frequente che è nato un framework specifico chiamato **ionic**

Ionic ~= Angular + Cordova

# Ionic CLI

Ionic fornisce il proprio tool, costruito intorno al CLI cordova, per la creazione di nuovi progetti, l'aggiunta di plugins, etc.

```
$ npm install -g ionic cordova
```

Creazione di un nuovo progetto:

```
$ ionic start <nomeprogetto> <template>
```

# Starter templates

<code>tabs</code>	A starting project with a simple tabbed interface
<code>blank</code>	A blank starter project
<code>sidemenu</code>	A starting project with a side menu with navigation in the content area
<code>super</code>	A starting project complete with pre-built pages, providers and best practices for Ionic development.
<code>conference</code>	A project that demonstrates a realworld application
<code>tutorial</code>	A tutorial based project that goes along with the Ionic documentation
<code>aws</code>	AWS Mobile Hub Starter



# Serving & deploy

Ionic può servire automaticamente l'applicazione nel browser host (sistema analogo al comando angular ng serve):

```
$ ionic serve
```

Il deploy dell'applicazione utilizza cordova direttamente:

```
$ ionic cordova platform add android
```

```
$ ionic cordova build
```

```
$ ionic cordova run
```

# Struttura del progetto

I sorgenti dell'applicazione Ionic vengono inseriti nella directory `src/` e compilati automaticamente nella dir `www/` durante il deploy dell'applicazione

Meccanismo analogo in cui abbiamo precedentemente configurato l'app Angular per il sistema mobile

# Componenti

Ionic fornisce dei componenti (Angular) built-in per sviluppare velocemente l'applicazione utilizzando elementi di layout tipici dei sistemi operativi mobile

<https://ionicframework.com/docs/components/>

Alerts, Badges, DateTime, List, Toast, etc.

# Navigation

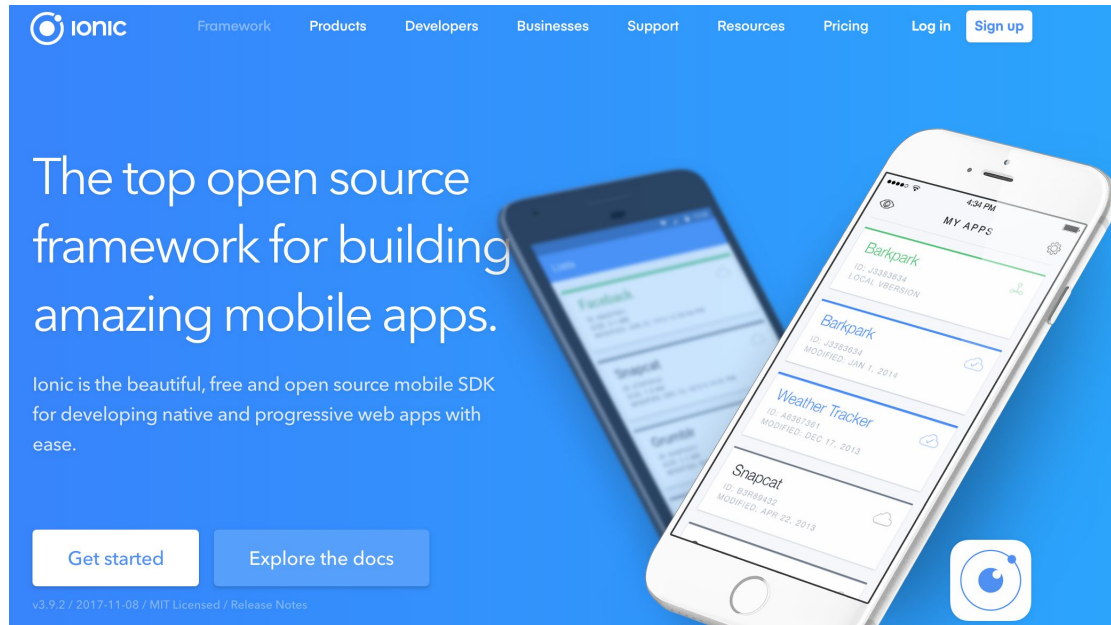
Ionic comprende un sistema per “navigare” tra pagine diverse secondo un modello a stack.

Il componente `<ion-nav>` si comporta in modo simile al componente `<router-outlet>` di Angular.

`<ion-nav>` è la direttiva esportata dal componente NavController, che può essere iniettato in ciascun componente che deve modificare programmaticamente la pagina da visualizzare

# Per saperne di più...

<https://ionicframework.com/framework>



The screenshot shows the Ionic Framework website homepage. The header features the Ionic logo and navigation links: Framework, Products, Developers, Businesses, Support, Resources, Pricing, Log in, and Sign up. The main content area has a blue background with the text "The top open source framework for building amazing mobile apps." and a description of Ionic as a free and open source mobile SDK. Two smartphones are displayed, showing app interfaces. At the bottom, there are buttons for "Get started" and "Explore the docs", and a footer with version and license information.

ionic

Framework Products Developers Businesses Support Resources Pricing Log in Sign up

The top open source framework for building amazing mobile apps.

Ionic is the beautiful, free and open source mobile SDK for developing native and progressive web apps with ease.

Get started Explore the docs

v3.9.2 / 2017-11-08 / MIT Licensed / Release Notes