

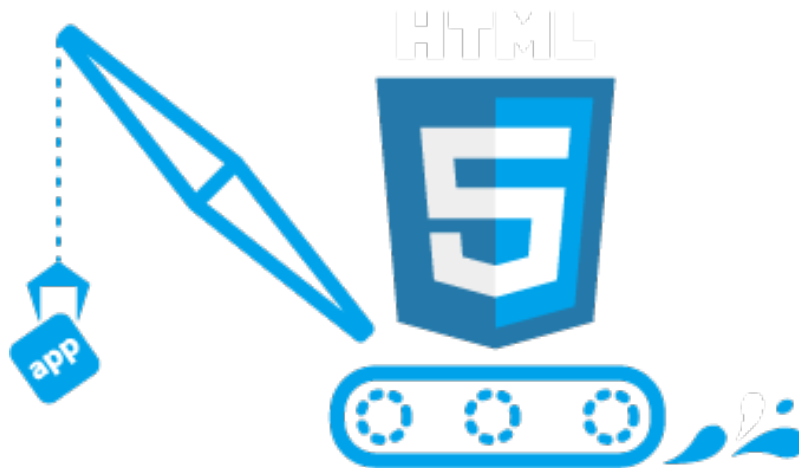


We create innovate software products that appeal to global audiences

Hackathon 2014

Javascript Storage & HTML5 Mobile Apps

Octubre 2014



Hackathon 2014

Javascript Objects

Octubre 2014





Javascript - Tipos de Datos

Undefined

undefined

Null

null

Boolean

true

String

"hello"

Number

3.141516

Object

{ name : "Bernardo" }

Function

function calc() { ... }

Array

[1 , 2 , 3 , "Some Text"]

RegExp

/pattern/



Javascript - Object

Definición de un objeto

```
var myObject = {  
  a: undefined,  
  b: null,  
  c: true,  
  pi: 3.141516,  
  name: "Bernardo",  
  available: true,  
  fn: function bar() {},  
  obj: {val: 123}  
};
```



Javascript - Primitive vs Object

Primitivos - Copia por Valor

```
var name1, name2;  
name1 = "Bernardo";  
name2 = nombre1;  
  
/*  
 * name1 = "Bernardo"  
 * name2 = "Bernardo"  
 */  
  
name2 = "Nick";  
  
/*  
 * name1 = "Bernardo"  
 * name2 = "Nick"  
 */
```

Objetos - Copia por Referencia

```
var object1, object2;  
  
object1 = {  
    name: "Bernardo"  
};  
  
object2 = object1;  
  
object2.name = "Nick";  
  
/*  
 * object1.name = "Nick"  
 * object2.name = "Nick"  
 */
```



Javascript - The Key! Functions are Objects

La definición de una función es en realidad la creación de un objeto

```
function myFunction(a,b){  
    return a+b;  
}
```

```
myFunction.newVar = "some text";
```

```
/*  
 * myfunction = {  
 *     name: "myFunction",  
 *     length: 2,  
 *     newVar: "some text",  
 *     prototype: ...   ← The Key  
 * }  
*/
```



Javascript - Object Methods

Como definir métodos dentro de un objeto

```
var myObject = {  
  id: 33,  
  getId: function(){  
    return this.id;  
  }  
};
```

```
/*  
 * myObject = {  
 *     id: 33,  
 *     getId: {name:"getId"....,prototype: ...},  
 * }  
 */
```




Javascript - Prototype ?

Prototype es la forma de herencia que utiliza javascript

```
var objectType = {  
  id: 33,  
  getId: function(){  
    return this.id;  
  }  
};  
var object2 = {  
  id: 33,  
  getId: function(){  
    return this.id;  
  }  
};
```

```
var objectType = {  
  id: 0,  
  getId: function(){  
    return this.id;  
  }  
};  
var child = Object.create(objectType);  
child.id = 2;  
  
var grandChild = Object.create(child);  
/**  
 * objectType.get() = 0,  
 * child.get()      = 2,  
 * grandChild.get() = 2  
 */
```



Javascript - Prototype Default Properties

Cualquier objeto que definamos va a incluir esta lista de métodos

```
var child
child.id = 1

var grandChild
/**
 * object
 * child
 * grandChild
 */
```

• getId(): Number	hackaton-fragments.js
■ id: Number	hackaton-fragments.js
■ var	hackaton-fragments.js
■ constructor: Function	JS Platform
• create(0: Object): Object	JS Platform
• defineProperties(0: Object, Properties: ...)	JS Platform
• defineProperty(0: Object, P: String, Att...)	JS Platform
• freeze(0: Object): Object	JS Platform
• getOwnPropertyDescriptor(0: Object, P: S...	JS Platform
• getOwnPropertyNames(0: Object): Array	JS Platform
• getPrototypeOf(0: Object): Object	JS Platform
• hasOwnProperty(V: String): Boolean	JS Platform
• isExtensible(0: Object): Boolean	JS Platform
• isFrozen(0: Object): Boolean	JS Platform
• isPrototypeOf(V: Object): Boolean	JS Platform
• isSealed(0: Object): Boolean	JS Platform
• keys(0: Object): Array	JS Platform
• preventExtensions(0: Object): Object	JS Platform
• propertyIsEnumerable(V: String): Boolean	JS Platform
• seal(0: Object): Object	JS Platform
• toLocaleString(): String	JS Platform
• toString(): String	JS Platform
• valueOf(): Object	JS Platform
■ prototype	JS Platform

objectType.



Javascript - Let's Organize Classes & Data

Usamos Prototype Model para organizar métodos y datos

```
var moviePrototype = {
  constructor: function(code,name){
    this._imdbCode = code;
    this._name = name;
  },
  getImdbCode : function(){
    return this._imdbCode;
  },
  getName: function(){
    return this._name;
  }
};

var movie1 = Object.create(moviePrototype);
movie1.constructor(1123,"Scary Movie");

var movie2 = Object.create(moviePrototype);
movie2.constructor(3324,"Forrest Gump");

movie1.getImdbCode(); // 1123
movie2.getName(); // "Forrest Gump"
```



Javascript - Let's Organize Classes & Data

Combinamos el prototipo de objeto con el prototipo de función para crear un modelo clásico de Clases

```
function MoviePrototype (code,name) {  
    this._imdbCode = code;  
    this._name = name;  
};  
MoviePrototype.prototype.getImdbCode = function(){  
    return this._imdbCode;  
};  
MoviePrototype.prototype.getName = function(){  
    return this._name;  
};  
  
var movie1 = new MoviePrototype(1123,"Scary Movie");  
var movie2 = new MoviePrototype(3324,"Forrest Gump");  
  
movie1.getImdbCode(); // 1123  
movie2.getName(); // "Forrest Gump"
```

Hackathon 2014

Javascript Storage Tools

Firebase

Octubre 2014





What's Firebase?



Firebase

Es uno de los tantos servicios de almacenamiento en la nube que podemos utilizar para almacenar nuestros objetos en javascript



```
var movie1 = new MoviePrototype(1123,"Scary Movie");  
var movie2 = new MoviePrototype(3324,"Forrest Gump");
```



How To



Firebase

Funcionamiento básico de Firebase

```
var dataSource = new Firebase('https://glb-hackton.firebaseio.com/');
```

```
//Algunos Eventos basicos disparados por el motor de Firebase
```

```
[-] dataSource.on('child_added',function(){});  
[-] dataSource.on('child_removed',function(){});  
[-] dataSource.on('child_changed',function(){});
```

```
//Agregar contenido
```

```
[-] dataSource.push({});
```

```
//Obtener un objeto almacenado
```

```
dataSource.child(reference);
```

Hackathon 2014

Run Examples

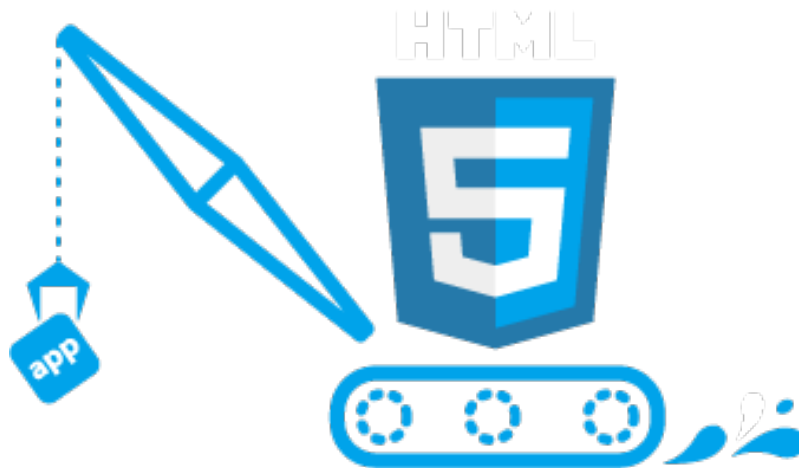
Octubre 2014



Hackathon 2014

HTML5 To App - Cordova

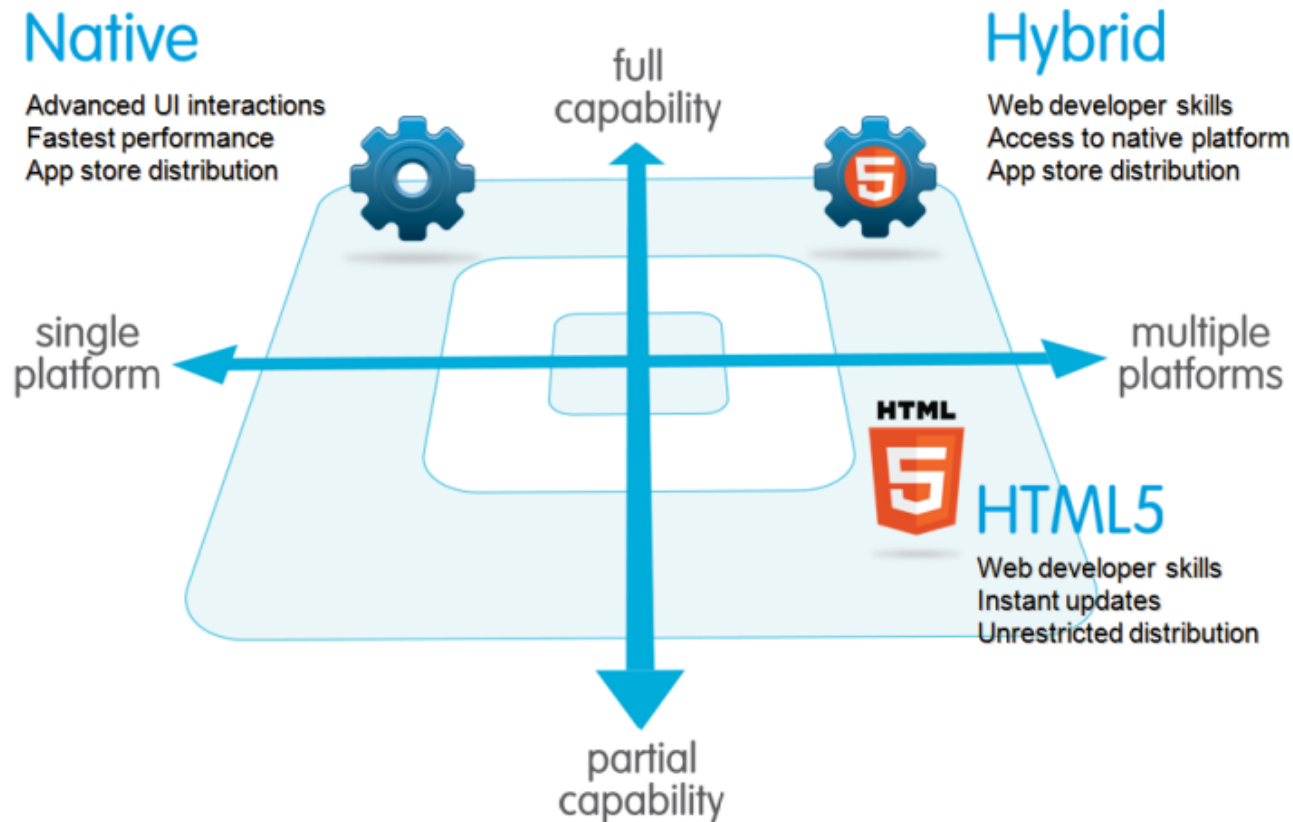
Octubre 2014





What does it mean?

Las aplicaciones Nativas en HTML5 como puerta de entrada al desarrollo móvil





Basics of HTML5 Native Apps

- Desarrollado en HTML5, javascript y css3
- Utilizan el browser del dispositivo como motor
- Utilizan una interfaz en javascript para comunicarse con el dispositivo



What's Apache Cordova

- Es una plataforma para el desarrollo de aplicaciones en html,css y javascript
- Provee una API en javascript para comunicarse con el dispositivo
- Permite portar la misma aplicaciones a múltiples plataformas haciendo ajustes mínimos o nulos.
- Soporta prácticamente todos los sistemas operativos móviles.

Android / iOS / Windows Phone / Blackberry / Mac OSX / Ubuntu / Windows / Firefox OS / Bada / Tizen / etc..



Can I play some music?

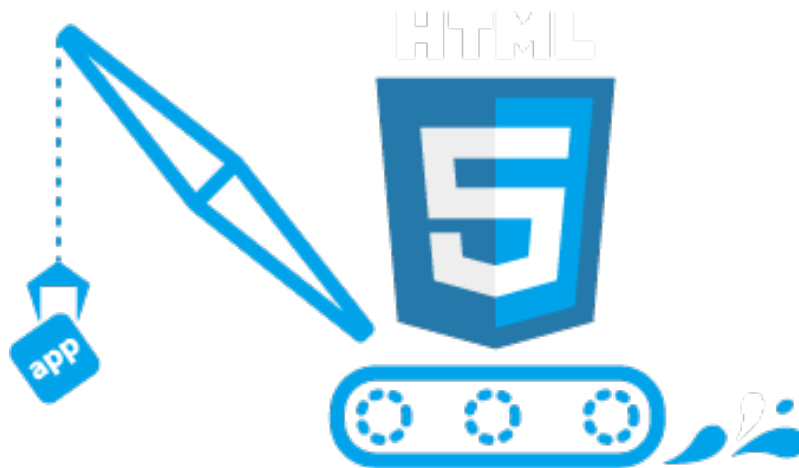
Lista de servicios del dispositivo que podemos aprovechar con cordova

- Battery Status
- Camera
- Console
- Contacts
- Device
- Device Motion
- Device Orientation
- Dialogs
- File
- File Transfer
- Geolocation
- Media
- Media Capture
- Network Information
- SplashScreen
- Statusbar
- Vibration

Hackathon 2014

Run Examples

Octubre 2014



➤ Thank you!