

Curso de especialización

Programación de  
Aplicaciones para iPhone/  
iPad de Apple Avanzada

**Primavera 2013 ( abril / mayo )**

Ángel Díaz

# Sesión 2 – Más allá de iOS

- PhoneGap
- Comunicación entre JS y iOS
- Subclasses y reutilización de código
- Organización de Proyectos iOS
- Código en Bloques



Curso de especialización

Programación de  
Aplicaciones para iPhone/  
iPad de Apple Avanzada



Phone**Gap**

# PhoneGap

Ángel Díaz

# PhoneGap (I)

- a.k.a. Cordova (CDV)
- ¿Qué es?
  - Framework
- ¿Quién está detrás?
  - **Adobe** ( compro Nitobi )
- **Marketing**
  - OpenSource
  - Gratuito
  - **Write Once, Deploy to Multiple Platforms**



# PhoneGap (II)



- **¿En qué se basa?**

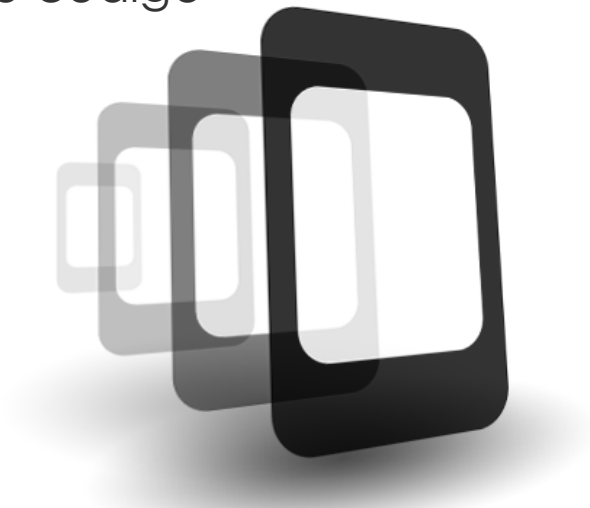
- Programación HTML5 + CSS3
- Javascript

- **¿Qué ofrece?**

- Acceso a Funcionalidades Nativas desde código Javascript

- **Beneficios**

- PhoneGap Plugins
- Comunidad de usuarios



PhoneGap

# PhoneGap (III)



- **¿Para que plataformas está disponible?**
  - iOS
  - Android
  - Windows Phone
  - BlackBerry
  - webOS
  - symbian
  - bada
  
- **Versión Actual**
  - 2.6.0
  
- **Actualizaciones**
  - Semanales



# PhoneGap (IV)

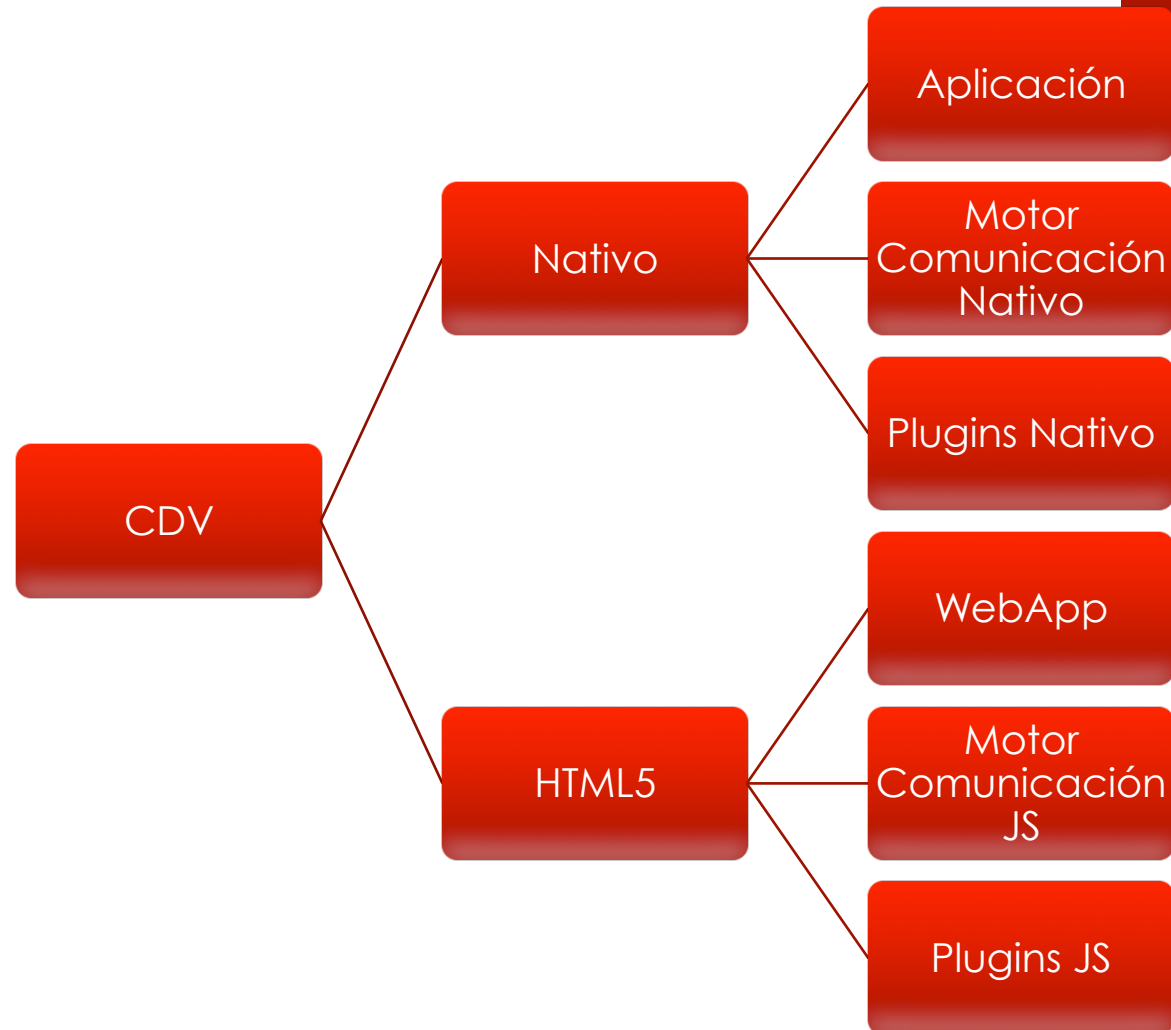
- Descarga
  - <http://phonegap.com/download>
- Funcionalidades soportadas por plataforma
  - <http://phonegap.com/about/features>
- Documentación
  - <http://docs.phonegap.com/en/2.6.0/index.html>
  - <https://github.com/phonegap/phonegap/wiki>
- Plugins
  - <http://github.com/phonegap/phonegap-plugins>



PhoneGap

# PhoneGap (V)

- ¿Que capas/partes tiene?





# PhoneGap (V)

- ¿Cómo funciona?



# PhoneGap (VI)

- Ejemplo PhoneGap/Cordova



# PhoneGap (VII)

- Ejemplo Uso Plugin PhoneGap/Cordova



Curso de especialización

Programación de  
Aplicaciones para iPhone/  
iPad de Apple Avanzada

**JS + iOS**

Ángel Díaz

# JS + iOS (I)

- Comunicación JS + iOS es uni-direccional
  - Por defecto en el SDK
    - UIWebView
      - `stringByEvaluatingJavaScriptFromString`
  
- Problema!
  - ¿Cómo podemos obtener feedback “complejo” de la capa JavaScript?
    - **NativeBridge**



# JS + iOS (II)

- **NativeBridge**

- URL de descarga:

- <https://github.com/ochameau/NativeBridge>

- ¿En qué consiste?

- WebView++
- JSON



# JS + iOS (III)

## ■ NativeBridge

- ¿Qué nos permite?
  - Callbacks JS vs. iOS
    - El javascript puede disparar eventos que tengan lugar en la capa Nativa
  - Manejar objetos “complejos”
    - Con JSON



# JS + iOS (III)

- **NativeBridge**

- ¿Cómo funciona?
  - WebView++
    - Carga de archivo
    - Carga de **mensajes**





# JS + iOS (III)

- Ejemplo NativeBridge



# JS + iOS (III)

- Ejemplo NativeBridge II



Curso de especialización

Programación de  
Aplicaciones para iPhone/  
iPad de Apple Avanzada

## Subclasses

Ángel Díaz

# Subclasses (I)

- Factor clave en la reutilización de código
- Ahorro de esfuerzos
- Abstracción
- Estandarización
- Rapidez



# Subclasses (II)

- iOS SDK ya es un buen ejemplo de reutilización de código y de subclaseado.
- Ejemplo:
  - UIButton, UILabel, ...
    - Subclases de UIView
    - Heredan el comportamiento gráfico de UIView pero cada tipo de datos tienen sus particularidades



# Subclasses (III)

## ■ Motivación

- Identificar en nuestras aplicaciones:
  - Partes
  - Componentes
  - Comportamientos
  - Clases
    - Que pueden ser sensibles a
      - Reutilizar
      - Generalizar
      - Especializar



# Subclasses (IV)

- **Ejemplo Subclasses**





Curso de especialización

Programación de  
Aplicaciones para iPhone/  
iPad de Apple Avanzada

## Organización Proyectos iOS

Ángel Díaz



# Org. Proyectos iOS (I)

- La organización es clave en cualquier proyecto software sea cual sea su magnitud
- Es importante identificar y diferenciar diferentes fases en todo proceso de desarrollo de software
- Existen herramientas software que nos ayudan
  - SVN, Git, ...
  - Redmine, Mantis, Trac, ...



# Org. Proyectos iOS (II)

- Fases
  - Requerimientos
  - Planificación
  - Analisis Requerimientos
  - Diseño
  - Programación
  - Testeo
  
- Secuencial? Lineal?



# Org. Proyectos iOS (III)



Curso de especialización

Programación de  
Aplicaciones para iPhone/  
iPad de Apple Avanzada

## Código en Bloques

Ángel Díaz

# Código en bloques (I)

- Síncrono vs. Asíncrono
  - Código Síncrono
    - Bloquea
    - Espera
    - Secuencial
  - Un solo hilo de ejecución
  - Debug Fácil
  - Mucho orden para reutilizar
- Delegados



# Código en bloques (II)

- Síncrono vs. Asíncrono
  - Código Asíncrono
    - No Bloquea
    - No Espera
    - No es Secuencial
  - Similar al concepto de threads
  - Debug Complejo
  - Reutilización Código Fácil
  - Optimización Código Fácil
- Sin Delegados



# Código en bloques (III)

- Síncrono vs. Asíncrono

- Código Síncrono

- (void) doAnotherThing {

- ...

- }

- (void) doSomething {

- ...

- }

- ...

- (void) mainRun {

- [self doSomething];

- [self doAnotherThing];

- }

- Código Asíncrono



# Código en bloques (IV)

- Síncrono vs. Asíncrono

- Código Asíncrono

- (void) doAnotherThing {  
    ...  
}
  - (void) doSomethingAndThenDo:((void (^)  
    (void)) aBlock {  
    ...  
}
  - ...
  - (void) mainRun {  
    [self doSomethingAndThenDo:^(  
        [self doAnotherThing];  
    )];  
}



# Código en bloques (V)

- Síncrono vs. Asíncrono
  - Código Asíncrono
    - Servicios Web
    - Operaciones con dependencia de otras
  - Ejemplos claros:
    - Animaciones UIView
    - AFNetworking
- Links:
  - [http://developer.apple.com/library/ios/#featuredarticles/Short\\_Practical\\_Guide\\_Blocks/index.html](http://developer.apple.com/library/ios/#featuredarticles/Short_Practical_Guide_Blocks/index.html)
  - [http://developer.apple.com/library/ios/#documentation/cocoa/Conceptual/Blocks/Articles/00\\_Introduction.html](http://developer.apple.com/library/ios/#documentation/cocoa/Conceptual/Blocks/Articles/00_Introduction.html)



# Código en Bloques (VI)

- Ejemplo Código en Bloques



Curso de especialización

Programación de  
Aplicaciones para iPhone/  
iPad de Apple Avanzada

**Literales**

Ángel Díaz

# Literales (I)

- Ayudan en:
  - **Declaración variables**
  - Traspaso de información entre tipos de datos
- Declaración tradicional:

```
NSNumber *aNumber = [NSNumber  
numberWithFloat:3.0f];
```

- Declaración con literales:

```
NSNumber *aNumber = @(3.0f);
```



# Literales (II)

- Declaración tradicional:

```
NSArray *aArray = [NSArray arrayWithObjects:  
@"obj1", @"obj2", @"obj3", nil];
```

- Declaración con literales:

```
NSArray *aArray = @[ @"obj1", @"obj2",  
@"obj3" ];
```



# Literales (III)

- Declaración tradicional:

```
NSMutableArray *aArray = [NSMutableArray  
arrayWithObjects: @"obj1", @"obj2", @"obj3",  
nil];
```

- Declaración con literales:

```
NSMutableArray *aArray = @[ @"obj1", @"obj2",  
@"obj3" ].mutableCopy;
```



# Literales (IV)

- Declaración tradicional:

```
NSDictionary *aDictionary = [NSDictionary  
dictionaryWithObjectsAndKeys: @"obj1",  
@"key1", @"obj2", @"key2", @"obj3", @"key3",  
nil];
```

- Declaración con literales:

```
NSDictionary *aDictionary = @{ @"key1" :  
@"obj1", @"key2" : @"obj2", @"key3" : @"obj3" };
```



Sesión 2

# Literales (V)



- Declaración tradicional:

```
NSMutableDictionary *aDictionary =  
[NSMutableDictionary  
dictionaryWithObjectsAndKeys: @"obj1",  
@"key1", @"obj2", @"key2", @"obj3", @"key3",  
nil];
```

- Declaración con literales:

```
NSMutableDictionary *aDictionary = @{ @"key1" :  
@"obj1", @"key2" : @"obj2", @"key3" :  
@"obj3" }.mutableCopy;
```



# Literales (VI)

- Links:

- <http://ijoshsmith.com/2012/07/29/objective-c-literals-for-ios-in-xcode-4-4/>
- <http://blog.bignerdranch.com/398-objective-c-literals-part-1/>



# Literales (VII)

- Ejemplo Literales

