

Seminario iOS. Cáceres. 17 Noviembre 2012



Fernando J. Pinilla
[@mendel81](#)

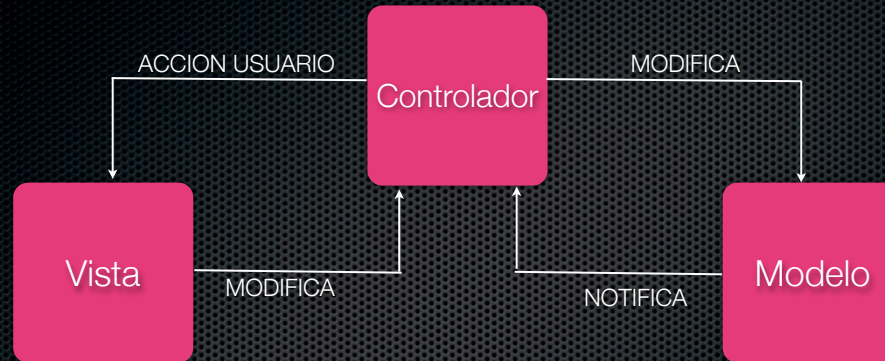
Indice

- Introducción a Objective-C y Cocoa Touch. Sintaxis y clases de Cocoa-Touch.
- Creación de un nuevo proyecto en Xcode
- Introducción a Xcode y a Interface Builder
- Hola Mundo! A compilar
- Introducción a TableViews y creación de una Lista en iOS
- Twitter y su API, juguemos con ella
- Graph de Facebook, vamos a conectar nuestro perfil.
- Retrospectiva, qué hemos hecho y qué podemos hacer.
- Preguntas

Introducción. Objective-C y Cocoa Touch

- Objective-C es una extensión de lenguaje C.
- Las apps iOS se desarrollan utilizando Objective-C y las librerías Cocoa-Touch.
- Cocoa Touch está escrito en Objective-C y es un conjunto de APIs que nos permiten programar las apps.

Introducción. Objective-C y Cocoa Touch



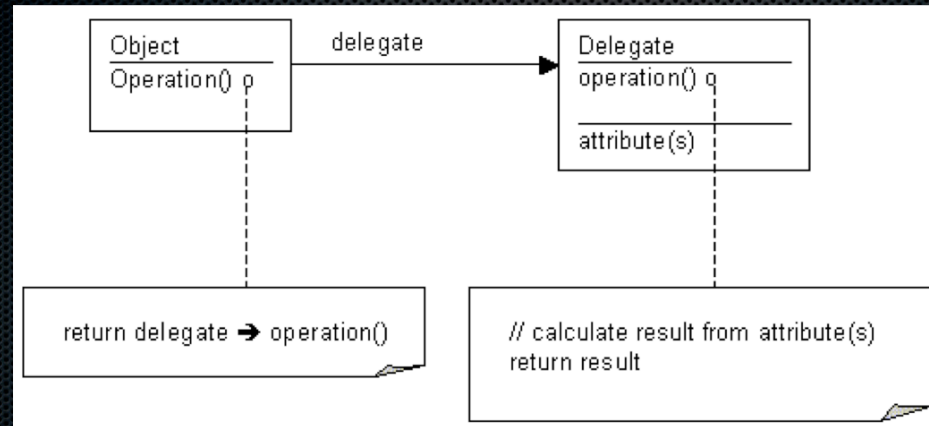
- - El modelo-vista-controlador es un patrón arquitectónico. Se emplea para estructurar las interfaces de los programas de una manera en la que separamos tareas diferentes en diferentes capas. Por un lado tenemos la presentación, por otro la lógica de aplicación y por otro la información del dominio.

Patrones básicos de Cocoa.

Delegación

- - El mecanismo de delegados consiste en que una clase A implemente los métodos de otra clase B para hacerse cargo del comportamiento de ésta, cuando esos métodos son llamados. El objeto A se llama delegado de B.
- Ej. UITableView delega el bucle para insertar las celdas de la tabla al UITableViewControllerDelegate

Patrones básicos de Cocoa. Delegación



Sintaxis Básica Objective-C

- Declaración de variables.
 - Ej. UILabel *etiqueta;
 - int numeroEntero;
 - NSString *cadena;

Sintaxis Básica Objective-C

■ Declaración de Métodos

- - (void)MostrarEtiqueta:(NSString *)texto;
- + (NSString *)getCadenaEtiqueta;
- - (void)setEntero:(int)numero;
- - (IBAction)cambiarVista:(id)sender;

Sintaxis Básica Objective-C

- Llamadas de métodos (mensajería)
- Ej. [MostrarEtiqueta:@"hola mundo!"];
- NSString *cadena = [self getCadenaEtiqueta];
- [self setEntero:25];

Sintaxis Básica Objective-C

- Instancias

- NSMutableArray *listaPersonas =
[NSMutableArray alloc];
- [listaPersonas init];

- Llamadas Anidadas

- NSMutableArray *listaPersonas =
[[NSMutableArray alloc] init];

Sintaxis Básica Objective-C

- Listas, arrays y diccionarios.
 - NSArray y NSMutableArray.
 - `[array objectAtIndex:x];`
 - `[array = @[objeto1, objeto2];`
 - `[array addObject: objeto3];`
 - NSDictionary y NSMutableDictionary.
 - `[diccionario objectForKey:@"nombre"];`
- Diccionario dentro de un Array:
 - `[[array objectAtIndex:x] objectForKey:@"nombre"];`

Sintaxis Básica Objective-C

- Bucles

- ```
for (int x=0; x < [listaPersonas count]; x++){
 NSLog(@"Nombre: %@", [listaPersonas
 ObjectAtIndex:x]);
}
```

- ```
if {} else {}
```

Gestión de Memoria

- La gestión de memoria se realiza como un contador.
- alloc = +1
- retain = +1
- release = -1
- autorelease
- dealloc
- Automático (ARC) (strong/weak)

Objetos en Cocoa Touch

▪ **UIView**

- `UIView *objeto = [[UIView alloc] initWithFrame:CGRectMake(0,0,300,300)];`
- Posición: `[objeto setCenter:CGPointMake(30,30)];`
- Tamaño: `[objeto setBounds:CGRectMake(0,0,300,300)];`
- Transparencia: `[objeto setAlpha:0.5];`
- Hidden: `[objeto setHidden:TRUE];`

Objetos en Cocoa Touch

▪ **UIView**

- Podemos tener varias vistas dentro de otras vistas de manera jerárquica.
- La mayoría de los objetos (UIImageView, UILabel, UIWebView, etc.) heredan de UIView y, por tanto, tienen las mismas propiedades que éstas.
- Diferencia entre UIView y UIViewController

Objetos en Cocoa Touch

▪ **UIViewController**

- `UIViewController *controlador = [[UIViewController alloc] initWithNibName:@"UIViewController" bundle:nil];`
- `[controlador.view addSubview: vista2];`
- `[controlador.view bringSubviewToFront: vista1];`

Objetos en Cocoa Touch

- **UILabel**

- **Hereda de UIView sus propiedades**

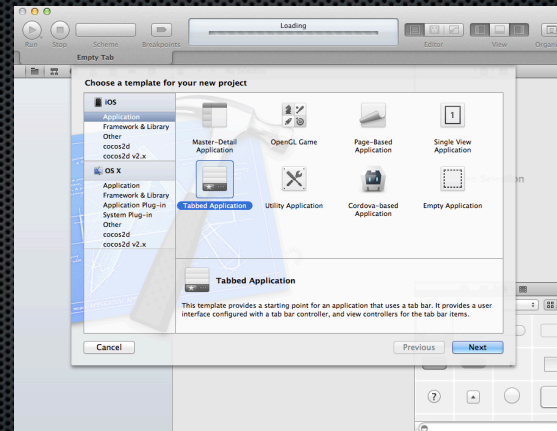
- UILabel *etiqueta = [[UILabel alloc]
initWithFrame:CGRectMake(0,0,300,300)];
- etiqueta.text = @"texto";
- [etiqueta setText:@"texto];

Salidas y Acciones

- IBOutlet (salida de Interface Builder)
- IBAction (acción)

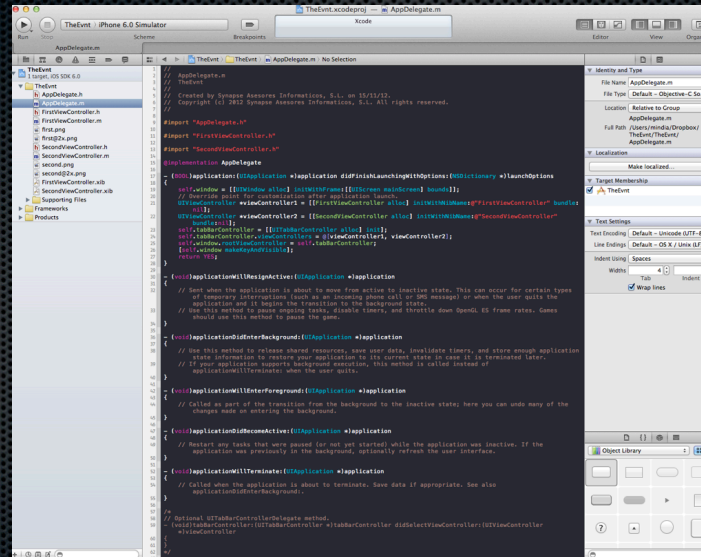
Nuevo Proyecto Xcode

■ Abrid Xcode

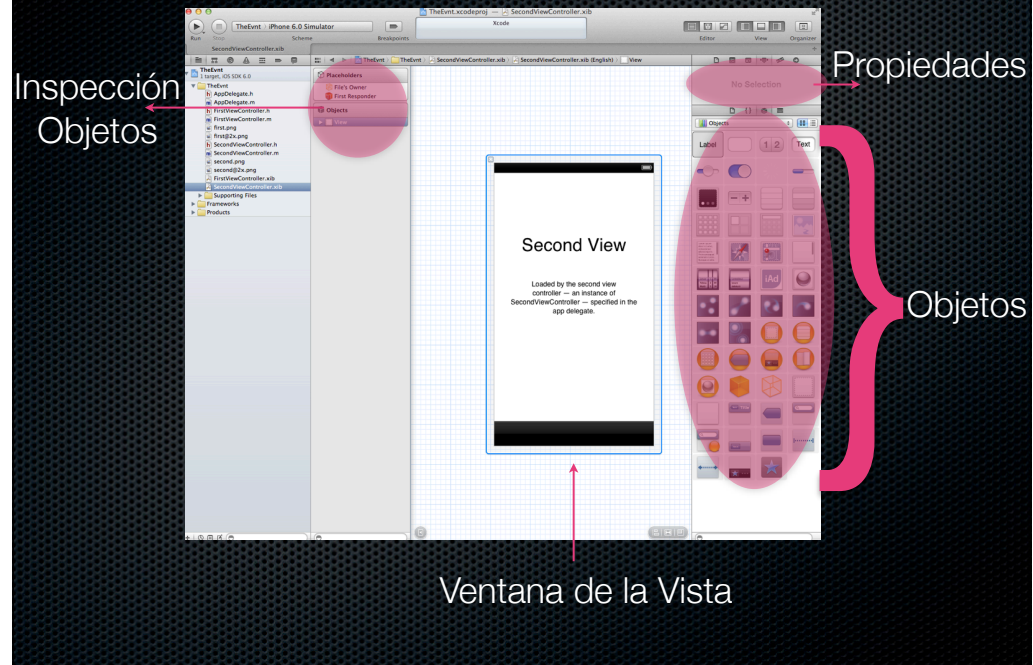


Xcode e Interface Builder

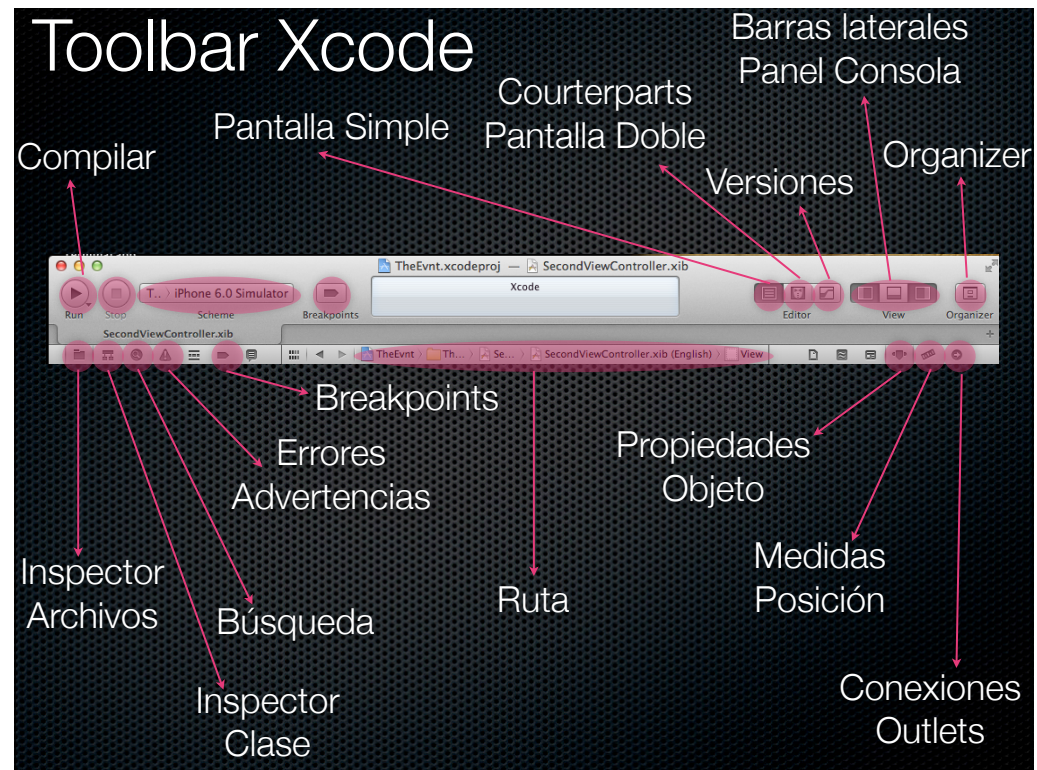
■ Xcode



Xcode e Interface Builder



Toolbar Xcode



UINavigationController

- Encargado de controlar el cambio de vistas (push, pop y back)

UITableView y UITableViewController

- - (NSInteger)tableView:(UITableView *)tableView
numberOfRowsInSection:(NSInteger)section
- - (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
- -(void)tableView:(UITableView *)tableView
didSelectRowAtIndexPath:(NSIndexPath *)indexPath

Blocks (bloques de código)

- Trozos de código que se pueden guardar en variables o pasar como argumentos
- También se pueden devolver como el resultado de un mensaje y ejecutar posteriormente

Blocks (bloques de código)

- Podemos usar bloques en las siguientes acciones:
 - Ordenación (el bloque se proporciona como método de comparación)
 - Notificación (cuando ocurra un evento ejecuta el bloque)
 - Gestores de error (cuando ocurra un error ejecutar bloque)
- Gestores de finalización (cuando termines de hacer una tarea, ejecutar bloque)
- Animación de vistas
- **Para multitarea mediante Grand Dispatch Central**

Api de Twitter y JSON

- https://api.twitter.com/1/statuses/user_timeline.json?include_entities=true&include_rts=true&screen_name=theevnt&count=25
- <http://search.twitter.com/search.json?q=%23iOSDev>

Parsing

- [NSJSONSerialization JSONObjectWithData:data options:kNilOptions error:&error]

Graph de Facebook

- <https://graph.facebook.com/mendel81>
- `NSData* data = [NSData dataWithContentsOfURL:
[NSURL URLWithString: @"https://
graph.facebook.com/mendel81"]];`
- `NSMutableArray *array = [[NSMutableArray alloc]
initWithArray:[NSJSONSerialization
JSONObjectWithData:data options:kNilOptions
error:&error]];`

Qué hemos hecho

- Perderle el miedo a Xcode y a Objective-c
- app con tabs y distintas vistas y controladores
- hemos interactuado con el usuario
- hemos cargado y controlado una tabla, arrays y diccionarios
- hemos conectado con webservices (Twitter y Facebook) y hemos parseado JSON y cargado los datos de forma asíncrona con hilos y bloques

Qué podemos hacer

- Si a esto añadimos notificaciones, persistencia de datos (Core Data o SQLite) y acceso a contactos del iPhone tendremos una app con funcionalidad similar a Whatsapp.
- Podemos realizar cualquier tipo de app con conexión a webservice y mantener los datos en la nube.
- Podemos crear la app de nuestra empresa o informativa en general.
- Podemos crear apps para interacción con el usuario

Preguntas y respuesta

- ✦ fernando@creapptiva.com
- ✦ @mendel81 (Twitter)

Gracias!

- Espero que no os hayáis dormido :-)