

1.6. iOS

Programación multimedia y dispositivos móviles
D.A.M. 2



CIFP A Carballeira

- Inicialmente llamado iPhone OS. Mediados de 2007.
- Desarrollado originalmente para el Iphone. Nueva definición del teléfono móvil.
- Más tarde en iPod Touch y actualmente en iPad.
- Actualizaciones desde la 1.x hasta la 1.1.5.
- La versión 1.0 incorporaba aplicaciones como Mail, Fotos, iPod, Calculadora, entre otras, presentes en las versiones actuales. No han sido modificadas prácticamente ni en sus interfaces ni en sus funcionalidades.

ios

- iPhone OS 2.0 (2008). Actualizaciones hasta la 2.2.
- Revolución de las aplicaciones móviles y uno de los modelos de negocio más productivos existente hoy en día.
- iPhone OS 3.0 (2009). Hasta la versión 3.1.3.
- Incluía el Spotlight (búsquedas en el dispositivo), posibilidad de incluir la API de Google Maps, operaciones de copiar/cortar/pegar, BlueTooth o P2P y librerías GPS.
- Versión soportada por los iPhones e iPads de primera generación.

ios

- iOS (2010). Versión 4.0.
- La versión 4.1.2 fue la última en los iPhone 3G y los iPod Touch de 2ª generación.
- iOS 5 (finales de 2011).
- Interfaz mejorada y funcionalidades como el asistente personal Siri, facilidades para sincronización sin cables, centro de notificaciones mejorado, servicio de iMessage, navegación web con pestañas, etc..
- Sólo soporta Apple TV (2ª generación), iPhone 3GS, iPhone 4, iPhone 4S, iPod Touch (3ª y 4ª generación) y iPad.

ios

- Se basa en capas.
 - Más altas: Servicios y tecnologías indispensables para el desarrollo de aplicaciones.
 - Más bajas: Encargadas de controlar los servicios básicos.



Arquitectura

- **Core OS:** Servicios de bajo nivel.
 - Archivos de memoria.
 - Manejo de memoria.
 - Seguridad.
 - Drivers del dispositivo.

Arquitectura

- **Core Services:** Servicios fundamentales del sistema utilizados por todas las aplicaciones.
- **Media:** Servicios gráficos y multimedia.

Arquitectura

- **Cocoa Touch:** Tres *frameworks*.
 - UIKit: Todas las clases necesarias para el desarrollo de la interfaz de usuario.
 - Foundation Framework: Clases básicas de acceso y manejo de objetos.
 - Servicios del S.O.
- Estas *frameworks* proporcionan la API de Cocoa para desarrollar aplicaciones.

Arquitectura

- Objective-C: Extensión orientada a objetos de la sintaxis de C.
- No existe recolector de basura, es recomendable estudiar en profundidad la documentación oficial de Apple sobre Objective-C y la gestión de memoria.
- En Objective-C 2.0 se incluye un recolector de basura (no soportado por iPhone). Cada instancia tiene un contador de referencias y cuando llega a cero el objeto es liberado de la memoria.

Lenguaje de programación

- Elementos comunes con el lenguaje C:
 - Sentencias de control de flujo.
 - Tipos de datos fundamentales.
 - Estructuras y punteros.
 - Conversiones implícitas y explícitas entre tipos.
 - Ámbito de las variables.
 - Funciones y su sintaxis.
 - Directivas del preprocesador.

Objective-C

- Las clases se crean mediante dos archivos:
 - *.h* para definirla.
 - *.m* para implementarla.
- Utilizar *#import* para indicar al compilador las clases que vamos a utilizar en nuestro código.

Objective-C

- Objective-C y C realizan procesos similares pero mejorados.
- Mayores diferencias con C++.
- Lenguaje muy limpio, pequeño y más rápido y fácil de aprender que C++.

Objective-C

mainapp.m

```
#import <UIKit/UIKit.h>
#import "SampleApp.h"
Int main(int argc, char **argv)
{
    NSAutoreleasePool *pool
        =[NSAutoreleasePool alloc init];
    Return UIApplicationMain(argc, argv,
        [SampleApp Class]);
}
```

Ejemplo

SampleApp.h

```
#import <CoreFoundation/CoreFoundation.h>
#import <Foundation/Foundation.h>
#import <UIKit/CDStructures.h>
#import <UIKit/UIWindow.h>
#import <UIKit/UIWView-Hierarchy.h>
#import <UIKit/UIHardware.h>
#import <UIKit/UIKit.h>
#import <UIKit/UIApplicacion.h>
#import <UIKit/UITextView.h>
#import <UIKit/UIView.h>
@interface SampleApp : UIApplication {
    UIView *mainView;
    UITextView *textView;
}
@end
```

Ejemplo

SampleApp.m

```
#import "SampleApp.h"
@implementation SampleApp
    - (void) applicationDidFininshLaunching: (id) unused
    {
        UIWindow *window;
        structCGRect rect 0 [UIHardware fullScreenApplicationContentRect];
        rect.origin.x = rect.origin.y = 0.0f;
        window = [[UIWindow alloc] initWithContentRect: rect];
        mainview = [[UIView alloc] iniwithFrame: rect];
        textView = [[UITextView alloc] iniWithFrame: CGRectMake(0.0f, 0.0f,
            320.0f, 480.0f)];
        [textView setEditable:YES];
        [textView setTextSize:14];
        [window orderFront: self];
        [window makeKey: self];
        [window_setHidden: NO];
        [window setContentView: mainView];
        [mainView addSubview:textView];
        [textView setText:@"Hello World"]
    }
}
```

Ejemplo

- Inicialmente a través de páginas o proyectos web.
- Apple vio la necesidad de controlar y unificar la programación de las aplicaciones para sus diferentes dispositivos.
- SDK (*Software Development Kit*, *Kit de Desarrollo de Software*) de Apple (2008). Conjunto de herramientas y tecnologías de desarrollo para crear aplicaciones para iPhone e iPod Touch.

Desarrollo para iOS

- Necesario inscribirse en el *iOS Dev Center*, donde se puede descargar gratuitamente.
- Para comercializarla: App Store. Requiere registrarse como desarrollador Apple y pagar una cuota.
- El desarrollador fija el precio y obtiene el 70% de las ventas realizadas a través del App Store.

Desarrollo para iOS

- Un dispositivo Apple solo ejecuta aplicaciones firmadas por Apple.
- Después de obtener el permiso de desarrollador, Apple expide un certificado que permite hacer pruebas en nuestro iPhone.

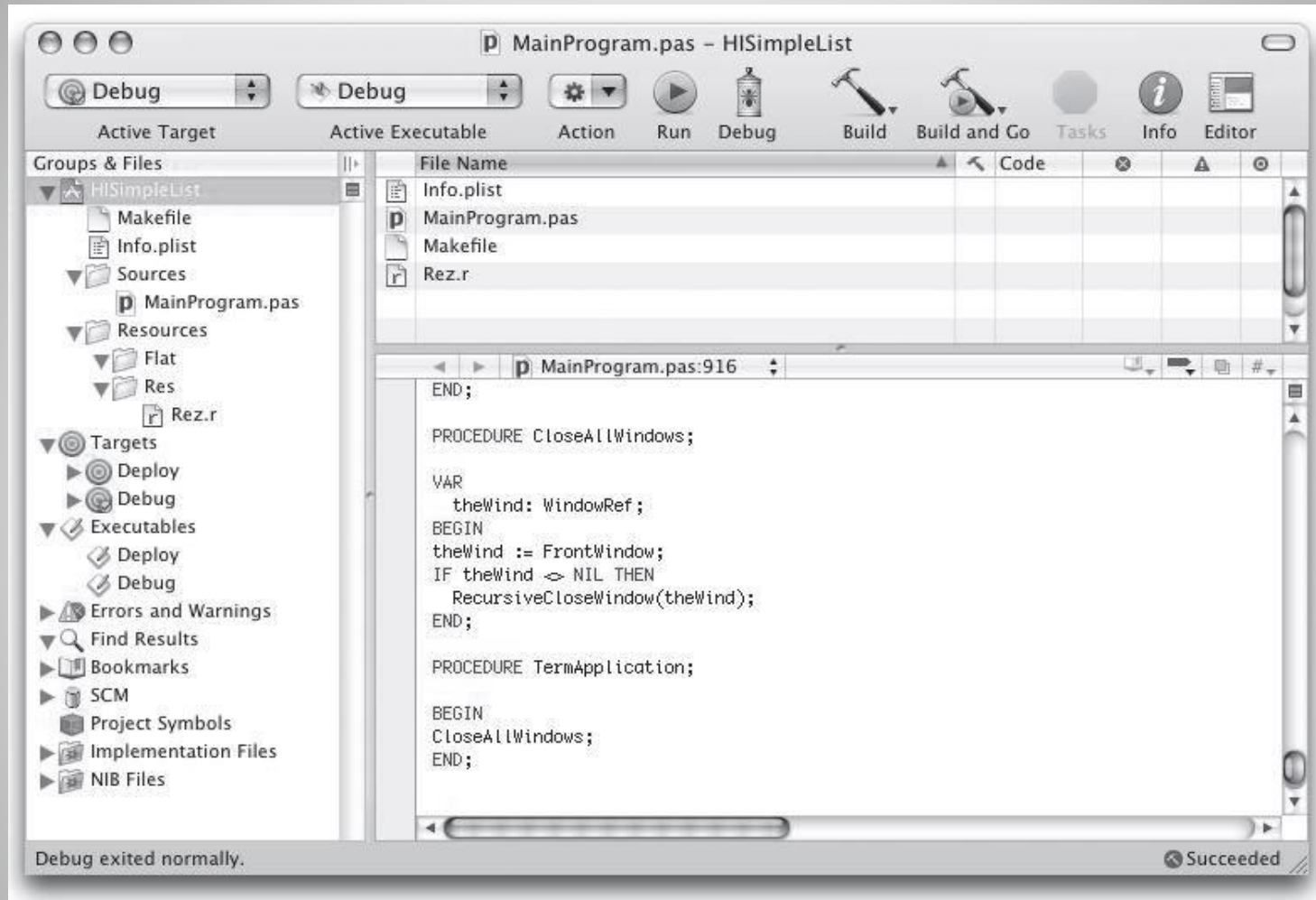
Desarrollo para iOS

- **Aplicación web:** Se ejecutan en un navegador y no hacen uso de las funcionalidades del dispositivo.
- **Aplicación nativa:** Utiliza las funcionalidades del dispositivo y se instala en el mismo, para acceder a ella no hace falta acceso a Internet.
- **Aplicación híbrida:** Combinan los dos tipos anteriores, puede instalarse en el dispositivo y además requiere de un navegador para utilizarla.

Aplicaciones iOS

- **Xcode:** IDE que permite escribir, compilar, ejecutar y depurar el código de la aplicación que se está desarrollando.
- Útil para organizar los archivos por proyectos y facilitar su importación/exportación.
- Simulador para probar la aplicación antes de ejecutarla en el dispositivo.
- Proporciona una serie de plantillas de aplicación.

Herramientas de desarrollo



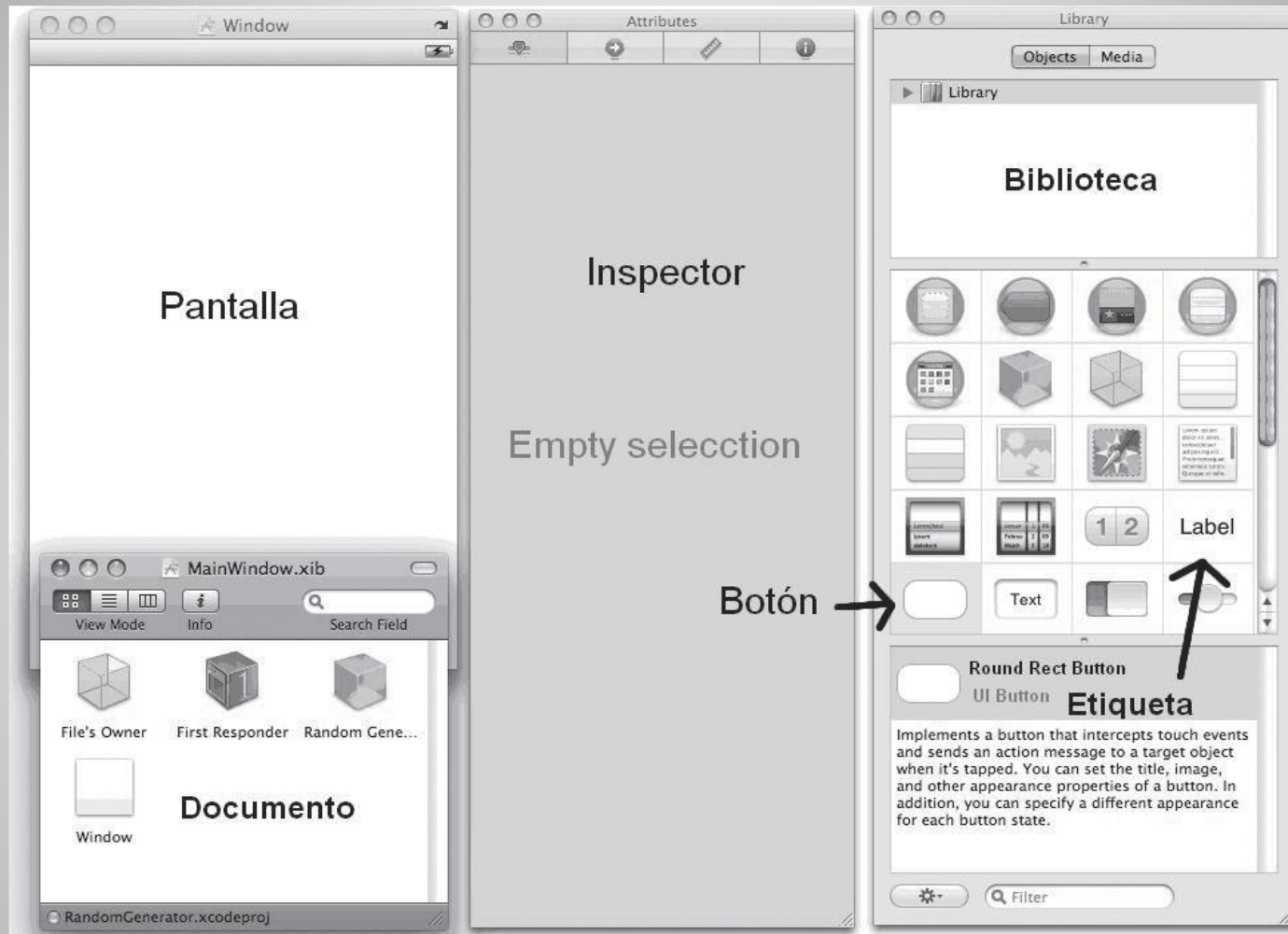
Xcode



CIFP A Carballeira

- **Interface Builder:** Permite diseñar la interfaz gráfica de la aplicación.
- Herramienta gráfica que trabaja en conjunto con Xcode.
- Desde la versión 4 de Xcode se puede programar el código y diseñar la interfaz en la misma ventana, arrastrando y soltando elementos como controles o componentes.
- Archivos generados: Extensión *.nib*.

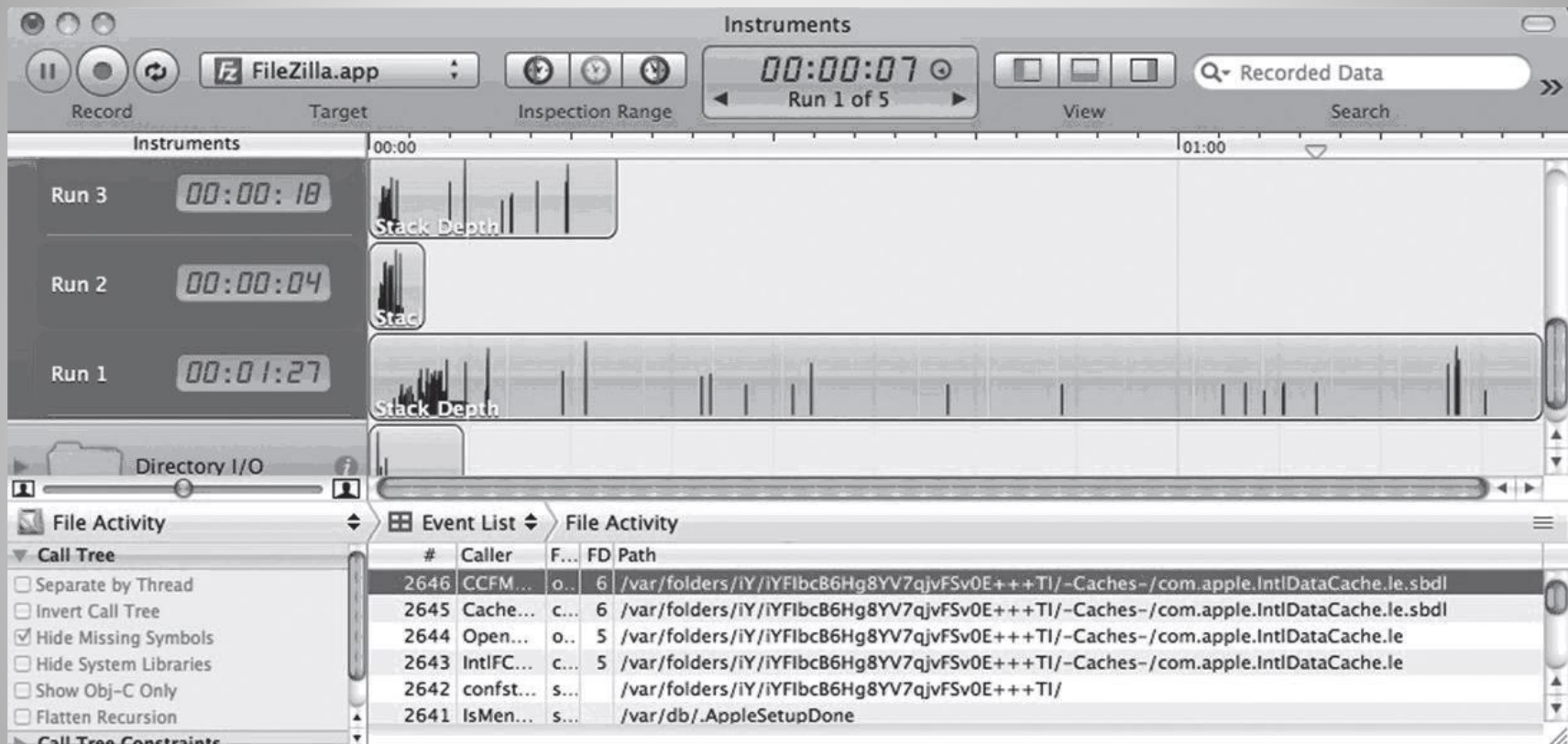
Herramientas de desarrollo



Interface Builder

- **Instruments:** Permite analizar el comportamiento de la aplicación mediante el análisis de parámetros como:
 - Memoria consumida por la aplicación.
 - Ancho de banda consumido.
 - Recursos utilizados.
 - Energía requerida.
 - Datos acerca de las animaciones.
 - Actividad relacionada con archivos y *sockets*.

Herramientas de desarrollo



Instruments

- **Funcionalidades:**

- Rotación, cuando gira el dispositivo la aplicación rota con él.
- Cambio de dispositivo, permite comprobar el funcionamiento tanto en un iPhone como en un iPad.
- Comprobar el funcionamiento de la aplicación para diferentes versiones del iOS.
- Avisos de memoria, permite comprobar el funcionamiento de la aplicación dependiendo del consumo de recursos.

Emuladores





Emuladores