

# Sistemas Operativos Móviles: iOS

Victor Tortolero, 24.569.609

Sistemas Operativos, FACYT

24 de julio de 2016

iOS fue presentado por primera vez por Apple en conjunto con el iPhone de primera generación en Enero del 2007. Cuando fue presentado el sistema operativo no contaba con un nombre oficial, un año mas tarde se presento como iPhone OS, y también se presento el SDK en conjunto con la version 2.0 de iPhone OS. En esta versión, se introdujo la App Store.

No fue sino hasta la versión 4.0, que obtuvo como nombre oficial iOS. También con esta versión se introdujo el multitasking. En la versión 5.0 se introdujo a Siri.



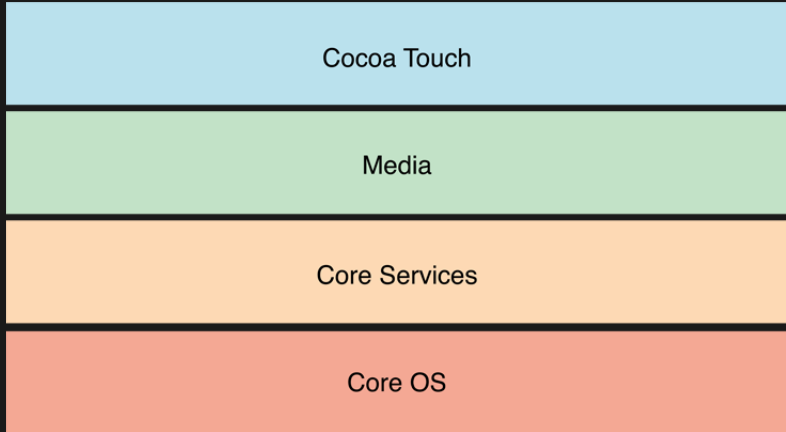
En la versión 7.0 hubo un total rediseño de la interfaz, todo enfocado a una mejor UX (Experiencia de usuario). A partir de la versión 8.0, los aplicaciones ahora pueden intercambiar información e integrarse unas con otras de nuevas maneras. La versión 9.0, agrega nuevas características al multitasking.



iOS usa una arquitectura por capas, cada capa usa un interfaz provista por la capa anterior a ella. Las capas mas bajas contienen servicios y tecnologías fundamentales. Las capas mas altas usan las abstracciones e interfaces provistas por las capas mas bajas y proveen servicios mas sofisticados. Cada capa contiene un conjunto de frameworks que son usados por la capa superior.

# iOS

## Arquitectura



- **Cocoa Touch:** Contiene frameworks para construir aplicaciones para iOS. Incluye soporte para el multitarea, entrada touch, notificaciones y otros servicios de alto nivel.
- **Media:** Contiene los servicios e interfaces necesarias para video, graficos, y audio. Presenta varias clases por las cual facilita el mostrar o generar contenido multimedia.

- **Core Services:** Contiene servicios fundamentales para las aplicaciones, como los servicios de localización, iCloud, entre otros.
- **Core OS:** Contiene todas las características de bajo nivel usadas por las demás capas. Maneja servicios fundamentales del sistema operativo como los hilos, administración de memoria, manejo del sistema de archivos.



El algoritmo de planificacion, es uno basado en colas multinivel con prioridades, cuyas prioridades estan divididas en 4 clases:

Prioridad	Características
Normal	Hilos con prioridad no importante
Alta	Prioridad mayor a la normal
Modo Kernel	Hilos que fueron creados dentro del kernel
Tiempo Real	Deben ser completados en un tiempo.

iOS nos brinda y utiliza varios mecanismos de sincronización, entre algunos tenemos:

- **Atomic Operations**
- **Locks**
- **Condition Variables**

**Atomic Operations:** Contiene servicios fundamentales para las aplicaciones, como los servicios de localización, iCloud, entre otros. Normalmente se usan para operaciones simples o cortas.

Function	Description
<code>OSIncrementAtomic(address);</code> <code>OSIncrementAtomic8/16/64(address);</code>	Adds 1 to the signed 8-, 16-, 32-, or 64-bit value at the specified address. The original value prior to the increment is returned.
<code>OSDecrementAtomic(address);</code> <code>OSDecrementAtomic8/16/64(address);</code>	Subtracts 1 from the signed 8-, 16-, 32-, or 64-bit value at the specified address. The original value prior to the decrement is returned.
<code>OSAddAtomic(amount, address);</code> <code>OSAddAtomic8/16/64(amount, address);</code>	Adds the value in “amount” to the signed 8-, 16-, 32-, or 64-bit value at the specified address. The original value prior to addition is returned.
<code>OSBitAndAtomic(mask, address);</code> <code>OSBitAndAtomic8/16(mask, address);</code>	Performs a bitwise AND operation of the value in “mask” and the 8-, 16-, or 32-bit unsigned value at the specified address. The original value prior to the bitwise operation is returned.
<code>OSBitOrAtomic(mask, address);</code> <code>OSBitOrAtomic8/16(mask, address);</code>	Performs a bitwise OR operation of the value in “mask” and the 8-, 16-, or 32-bit unsigned value at the specified address. The original value prior to the bitwise operation is returned.

**Locks:** Usados cuando se necesita sincronizar instrucciones o operaciones mas complejas y grandes. Cuando se pone un Lock alrededor de una sección del código, solo un hilo a la vez tendrá acceso a esta sección. Entre los tipos de locks que provee, tenemos Mutex, Recursive Lock, Read-Write Lock, Spin Lock, entre otros.

**Condition Variables:** Permite la sincronización entre múltiples hilos proveyendo un mecanismo por el cual un hilo puede suspender sus ejecución hasta que una condición en particular (o evento) se cumpla, cuando esto ocurre se manda una señal o mensaje a los hilos que esperan por esto.

El SDK de iOS y los frameworks que nos brindan las capas del sistema operativo, nos permiten realizar comunicación entre distintas aplicaciones de distintas formas, una es a través de Airdrop, o definir esquemas de URL personalizados para enviar información.

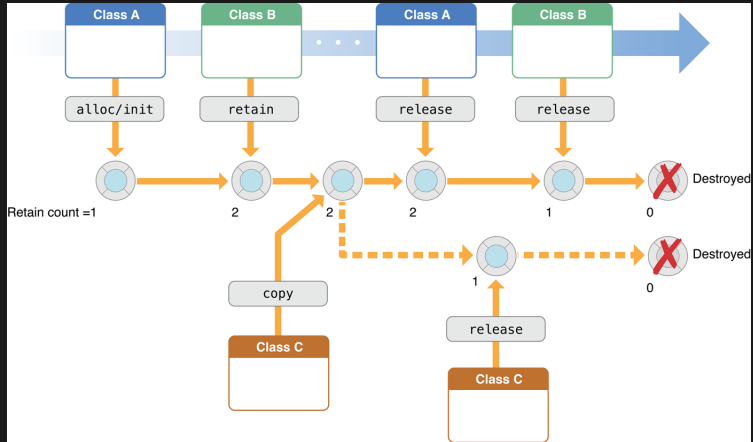
El kernel XNU sigue el estado de la memoria física usando una estructura `vm_page`. Existe una `vm_page` por cada página física en memoria. Las páginas disponibles forman parte de una de las siguientes listas:

- **Active List:** Contiene las páginas físicas que están asignadas aunque sea a un espacio de direcciones virtuales y han sido usadas recientemente.
- **Inactive List:** Contiene las páginas que están asignadas pero no han sido usadas recientemente.
- **Free List:** Contiene las páginas no asignadas.

El kernel enviara una señal al demonio pageout si detecta que el numero de paginas disponibles esta por debajo de un limite. En este caso se sacaran paginas de la lista de paginas no activas (Inactive List) con un algoritmo basado en LRU.



Cuando hablamos del desarrollo de apps para iOS, nos proveen dos metodos para el manejo de memoria en la aplicación.



- **MRR (Manual retain-release):** Explícitamente se maneja la memoria, llevando la cuenta de los objetos que se han creado.
- **ARC (Automatic Reference Counting):** No hace falta llamar a las funciones de manejo de memoria explícitamente, al compilar el código automáticamente se insertan las llamadas a las funciones de manejo de memoria necesarias. Tiene los beneficios de la Recolección de Basura pero sin los costos de rendimiento de la misma.

Mientras que la capa más alta de ese sistema operativo (Cocoa Touch) se encarga de manejar la entrada táctil, la capa de Media nos da acceso de alto nivel a la pantalla y el sonido, las capas más bajas nos brindan una interfaz más cercana al Hardware.

El sistema de archivos del kernel XNU se basa en el diseño de VFS (Virtual File System), el cual se caracteriza porque nos permite añadir fácilmente nuevos sistemas de ficheros, así como añadir capas a un sistema de archivo existentes (compresión, encriptación). Los archivos y carpetas son representadas por una estructura `vnode`. Existe un `vnode` por cada archivo activo en el kernel.

Tenemos que iOS soporta distintos sistemas de archivos, HFS+, HFS, UFS, NFS, SMB, UDF y AFP. El sistema de archivos primario de iOS es HFS+. Este sistema de archivos es robusto contra eventos como fallas de poder o fallos del kernel, la información puede ser traída al estado en donde se encontraba para tener al sistema de archivos en un estado consistente.

Y como el sistema de archivos de iOS está basado en el de UNIX, cada objeto del sistema de archivos tiene un conjunto de permisos de UNIX definidos por 3 atributos:

- **UID:** El id del usuario, el dueño del archivo.
- **GID:** El id del grupo.
- **UID:** Bits para indicar permisos y otros atributos.

iOS brinda ciertas características interesantes cuando hablamos de protección, por ejemplo permiten que el usuario remotamente con su cuenta de iCloud pueda localizar su dispositivo, revisar alguna información en este, e incluso borrar toda la información en caso de robo.

iOS encripta todos los archivos del sistema, para poder realizar un borrado rápido de toda la información en caso de que sea necesario.

En el desarrollo de iOS, Apple siempre pone primero al usuario, y por lo tanto versión a versión ha mejorado su UX (User Experience) de gran manera. En este sistema operativo la facilidad de uso para el usuario y la fluidez a la hora de realizar cualquier tarea son detalles clave.

También, al Apple desarrollar este OS para un Hardware limitado, han podido optimizarlo en gran manera.