

# Programación para Dispositivos Móviles(T.E.)

Ciclo 1 - 2019

Clase:003

Coordinador:Ing. César González



OBJECTIVE-C

# Unidad II

## Interfaz Grafica

Basado en

[https://developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/index.html#//apple\\_ref/doc/uid/TP40011343-CH2-SW1](https://developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/index.html#//apple_ref/doc/uid/TP40011343-CH2-SW1)

Otros materiales publicados años anteriores en el sitio de desarrolladores de apple

# Agenda

- **Desarrollo de Apps para IOS.**
  - **Pasos para el desarrollo de aplicaciones para IOS.**
  - **Introducción.**
  - **Estructuración.**
  - **Implementación.**
- **Diseño de interfaz humana.**
- **Lenguaje de desarrollo para IOS.**
- **Lineas Guia para un buen diseño de interfaz humana.**
- **Antes de iniciar el desarrollo de Apps para IOS**
- **Demo Xcode**



# Desarrollo de Apps para IOS

**Pasos para el desarrollo de aplicaciones para IOS son los siguientes:**

- 1 Inscribirse en el iOS Dev Center y descarga del SDK.
- 2 Registrarse como desarrollador de Apple (este paso es necesario solo para desarrolladores que quieran comercializar la aplicación desarrollada).
- 3 Elegir el tipo de aplicación a desarrollar.
- 4 Utilizar las herramientas de desarrollo para el diseño y puesta a punto de la aplicación.
- 5 Probar la aplicación, utilizando el simulador de iOS y luego sobre el dispositivo de Apple.
- 6 Distribuir la aplicación a través del App Store (una vez obtenida la firma de Apple).

# Desarrollo de Apps para IOS



## 1) Introducción

Al desarrollar aplicaciones, se utiliza el kit de desarrollo de software iOS (SDK) y Xcode, el entorno de desarrollo integrado de Apple (IDE). Xcode ofrece todo lo necesario para crear grandes aplicaciones para el iPhone, iPod touch y iPad. Incluye un editor de código fuente, un editor de interfaz gráfica de usuario, y muchas otras características. Xcode utiliza una sola ventana, llamada ventana de trabajo, que presenta la mayoría de las herramientas que necesita para desarrollar aplicaciones.

# Desarrollo de Apps para IOS



## 2. Estructuración.

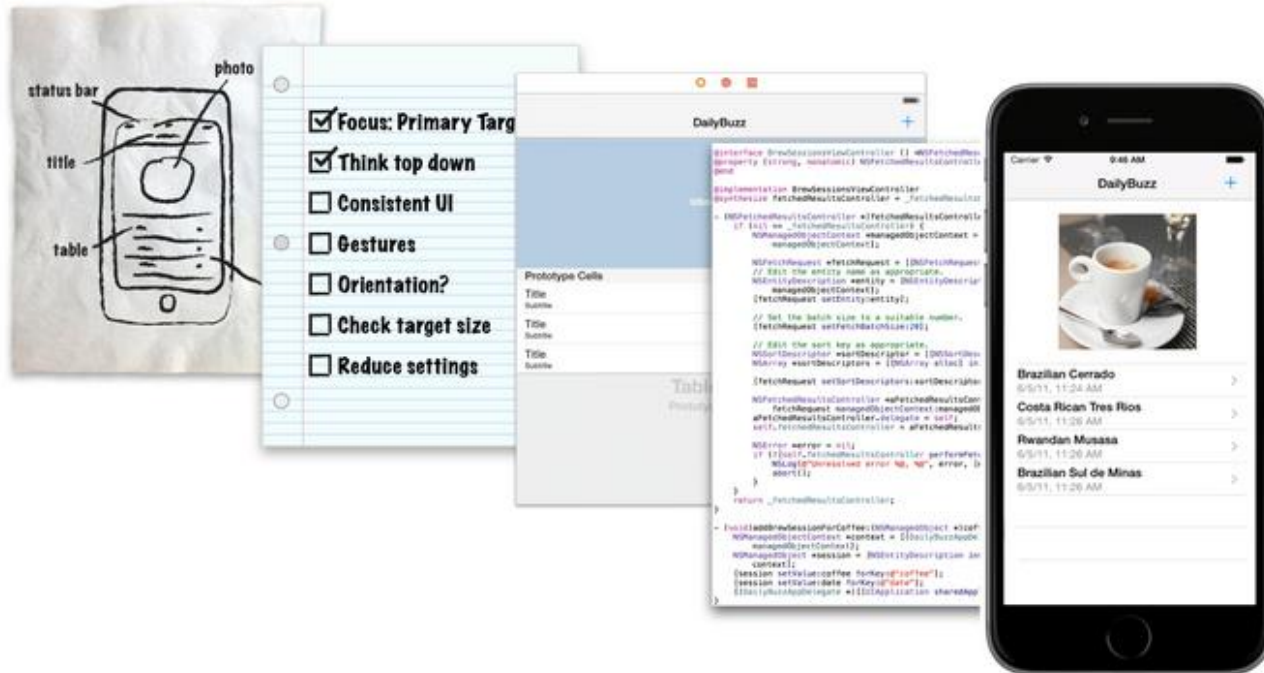
### 2.1) Definiendo el concepto

Quién es tu audiencia?

Cuál es el propósito de tu aplicación?

Cuál es el problema que tu aplicación trata de resolver?

Que tipo de contenido podrá tu aplicación incorporar?

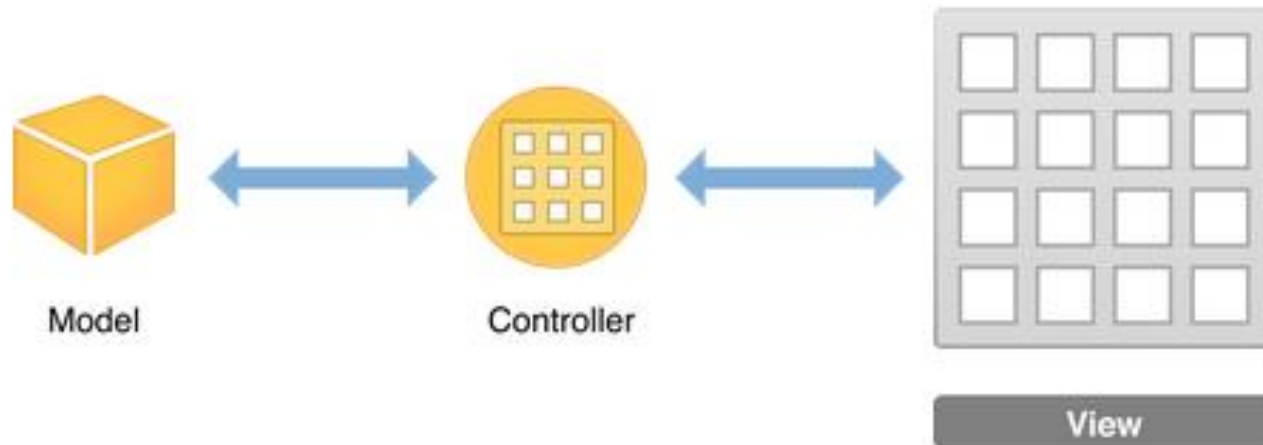


# Desarrollo de Apps para IOS



## 2.2) Diseñando la Interfaz de Usuario

Un usuario tiene que interactuar con una interfaz de aplicación en la forma más sencilla posible. Para diseñar la interfaz hay que tener al usuario en mente, y desarrollarla de forma eficiente, clara y sencilla.



# Desarrollo de Apps para IOS



## 2.3) Definiendo la interacción

Las aplicaciones de iOS se basan en la programación orientada a eventos. Es decir, el flujo de la aplicación se determina por eventos: eventos del sistema y las acciones del usuario. El usuario realiza acciones en la interfaz, que desencadenan los acontecimientos en la aplicación. Estos acontecimientos dan lugar a la ejecución de la lógica y la manipulación de los datos de la aplicación. La respuesta de la aplicación para la acción del usuario se refleja luego de vuelta en la interfaz. Debido a que el usuario, en lugar del desarrollador, tiene el control de cuándo ciertas partes del código de la aplicación se ejecutan, se debe identificar exactamente qué acciones puede realizar un usuario y lo que ocurre en respuesta a esas acciones.



# Desarrollo de Apps para IOS



## 2.3) Definiendo la interacción

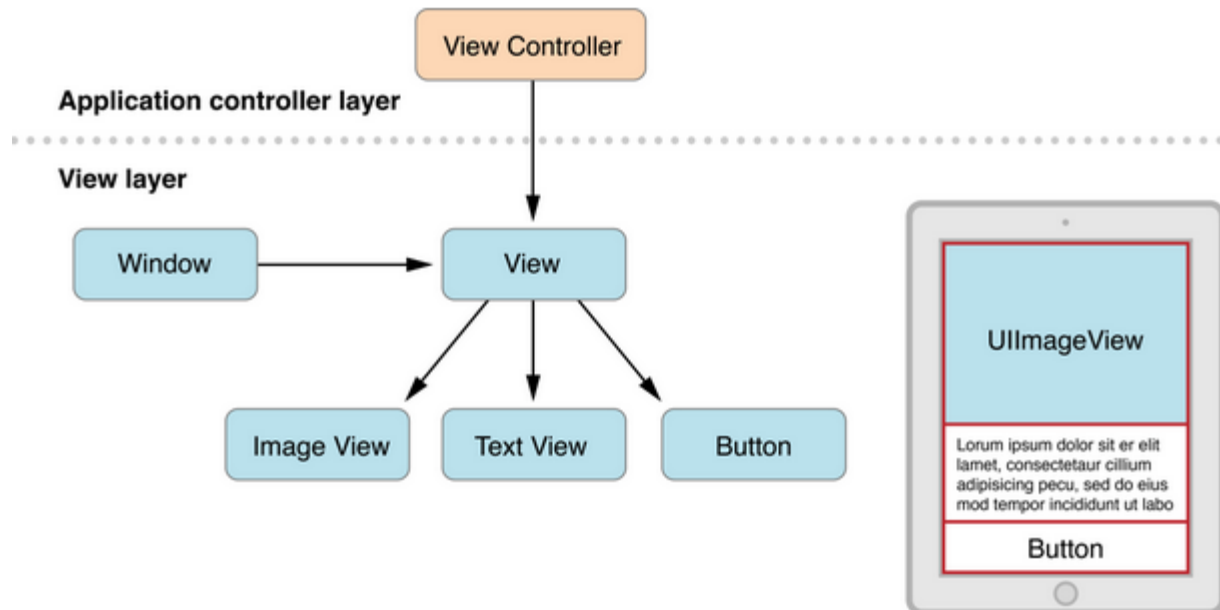
El programador define gran parte de la lógica de control de eventos en los controladores de vista. Los controladores son objetos que apoyan sus puntos de vista al responder a las acciones del usuario y llenar las opiniones con contenido. Objetos del controlador son un conducto a través del cual las vistas interactúan con el modelo de datos. Las vistas son notificadas de los cambios en los datos del modelo a través de los controladores de la aplicación, y los controladores comunican los cambios para iniciarlos por el usuario.

# Desarrollo de Apps para IOS



## 2.3) Definiendo la interacción(ViewController)

Después de construir una jerarquía básica de vistas, el siguiente paso es el control de los elementos visuales y responder a la entradas del usuario. En una aplicación para iOS, se utiliza un controlador de vistas (UIViewController) para gestionar una vista de contenido con su jerarquía de subvistas

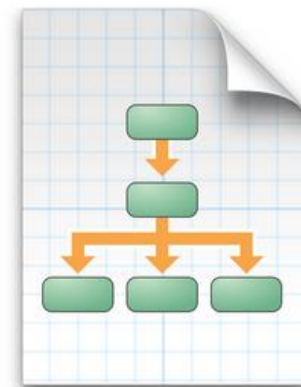
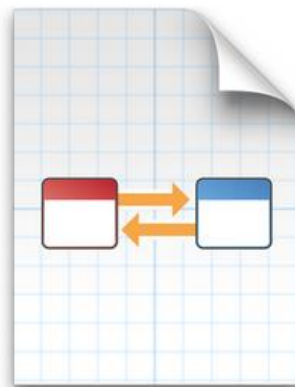
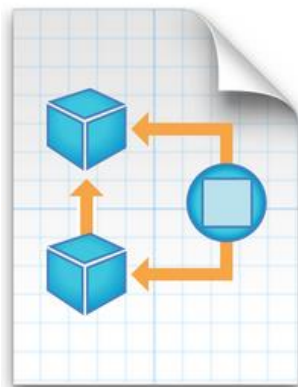


# Desarrollo de Apps para IOS



## 3) Implementación de una App (Patrones de Diseño)

Un patrón de diseño resuelve problemas comunes de ingeniería de software. Los patrones son diseños abstractos, no código. Se utilizan para ayudar a definir la estructura de modelos de datos y su interacción con el resto de la aplicación. Cuando se adopta un patrón de diseño, es para adaptar un modelo general a las necesidades específicas. No importa qué tipo de aplicación que está creando, es bueno conocer los patrones de diseño fundamentales utilizados en los frameworks. La comprensión de los patrones de diseño le ayuda a utilizar los frameworks de manera más eficaz y permite escribir aplicaciones que son más reutilizables, más extensible, y más fácil de cambiar.

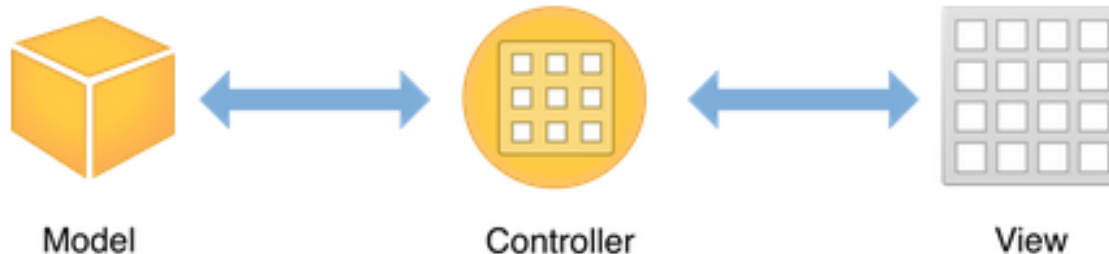


# Desarrollo de Apps para iOS



## 3.1) Patrón MVC

El Modelo-Vista-Controlador (MVC) es fundamental para un buen diseño para cualquier aplicación de iOS. MVC asigna los objetos en una aplicación para uno de los tres roles: modelo, vista o controlador. En este modelo, los **modelos** de seguimiento de los datos de su aplicación, **las vistas** muestran la interfaz de usuario y constituyen el contenido de una aplicación, y los **controladores** gestionan las vistas.



# Desarrollo de Apps para IOS



## 3.2) Patrón Target Action

Target-action es un diseño conceptualmente simple en el que un objeto envía un mensaje a otro objeto cuando se produce un evento específico. El mensaje **action** es un selector que se define en el código fuente, y el **target(objetivo)** es el objeto que recibe el mensaje, es un objeto capaz de llevar a cabo la acción, por lo general un View Controller. El objeto que envía el mensaje de acción es por lo general un control como un botón, control deslizante, o cambiar, que puede desencadenar un evento en respuesta a la interacción del usuario, como tocar, arrastrar, o cambiar de valor.

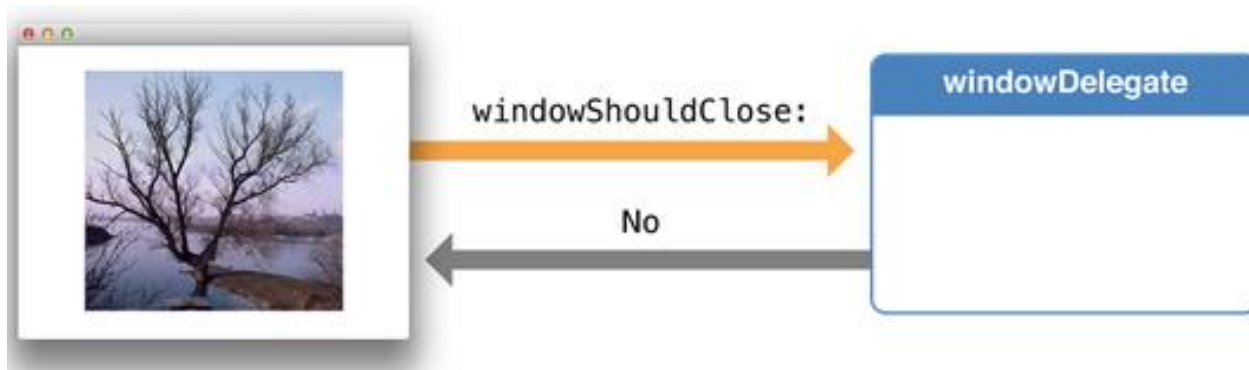


# Desarrollo de Apps para IOS



## 3.3) Patrón Delegation

El Delegation es un patrón simple y de gran alcance en el que un objeto en una aplicación actúa en nombre de, o en coordinación con otro objeto. El objeto delegar mantiene una referencia a otro objeto delegado y en el momento apropiado, el objeto delegar envía un mensaje al delegado. El mensaje informa al delegado de un evento que el objeto delegar está a punto de manejar o simplemente ha manejado. El delegado puede responder al mensaje mediante la actualización de la aparición (o estado) de sí mismo o de otros objetos en la aplicación, y en algunos casos se devolverá un valor que afecta a cómo se maneja un evento inminente.





# Desarrollo de Apps para iOS

El SDK de iOS amplía el conjunto de herramientas de Xcode para incluir las herramientas, los compiladores, y los marcos necesarios específicamente para iOS.

Mac App Store



Xcode FREE



Download





# Diseño de Interfaz Grafica (UI)

Hacer una aplicación de un concepto simple de la App Store no es una tarea fácil, el proceso y las buenas practicas son lo fundamental.







# Diseño de Interfaz Grafica (UI)



## 1) Diseño para IOS

**La deferencia.** La interfaz de usuario ayuda a la gente a entender e interactuar con el contenido, pero nunca compite con él.

**Claridad.** El texto es legible en todos los tamaños, los iconos son precisos y lúcidos, los adornos son sutiles y apropiados, y con un enfoque más preciso en la funcionalidad que motiva al diseño.

**Profundidad.** Capas visuales y movimiento realista imparten vitalidad y aumentan el placer y comprensión de la gente.



# Diseño de Interfaz Grafica (UI)

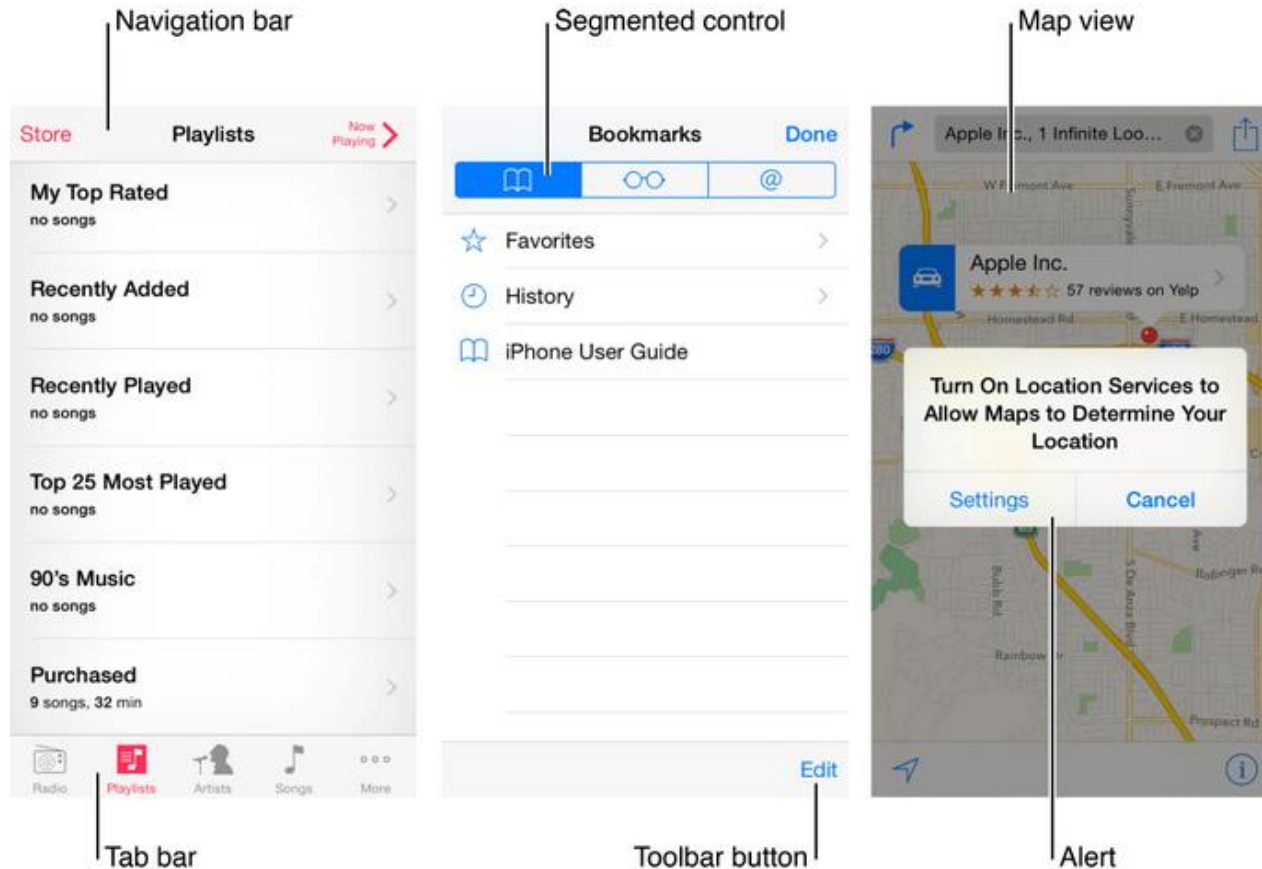
## 2) Anatomía para las Apps IOS

Casi todas las aplicaciones de iOS utilizan al menos algunos de los componentes de interfaz de usuario definidas por el marco UIKit. Conocer los nombres, funciones y capacidades de estos componentes básicos ayuda a tomar decisiones informadas al diseñar la interfaz de usuario de su aplicación.



# Diseño de Interfaz Grafica (UI)

## 2) Anatomía para las Apps IOS...

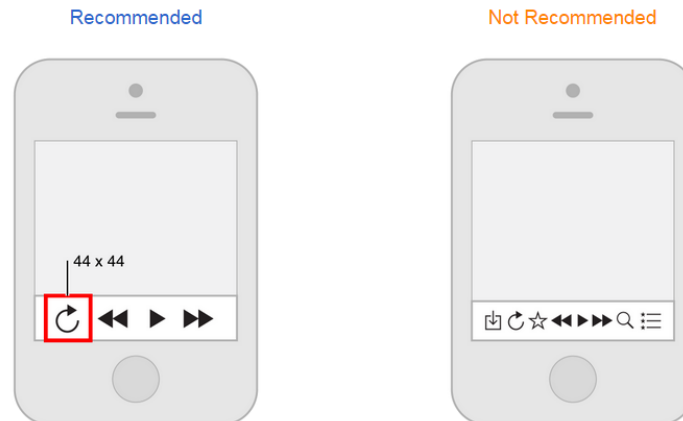




# Diseño de Interfaz Grafica (UI)

## 3) Adaptabilidad y Diseño

La gente en general quieren utilizar sus aplicaciones favoritas en todos sus dispositivos y en cualquier orientación. En iOS 8 y versiones posteriores, utiliza clases de tamaño y Auto Layout para ayudarle a cumplir con esta expectativa por la definición de cómo el diseño de pantallas, los ViewControllers y los View deben adaptarse al modificarse el entorno de visualización.





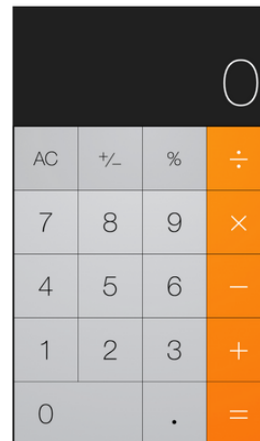
# Diseño de Interfaz Grafica (UI)

## 4) Inicio y detención

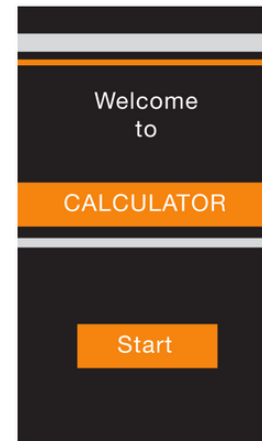
### Iniciar de forma instantánea

A menudo se dice que la gente pasa no más de un minuto o dos evaluar una nueva aplicación. Cuando usted hace la mayor parte de este breve período, presentando un contenido útil de inmediato, esto hace despertar el interés de los nuevos usuarios y dar a todos los usuarios una experiencia superior.

Recommended



Not Recommended





# Diseño de Interfaz Grafica (UI)

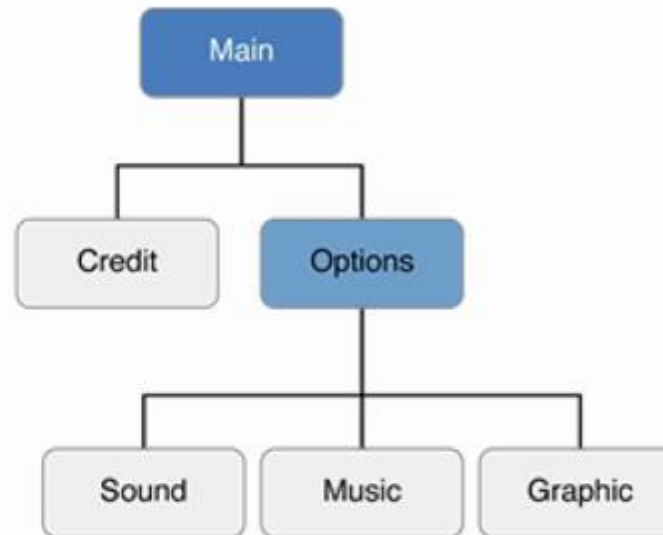
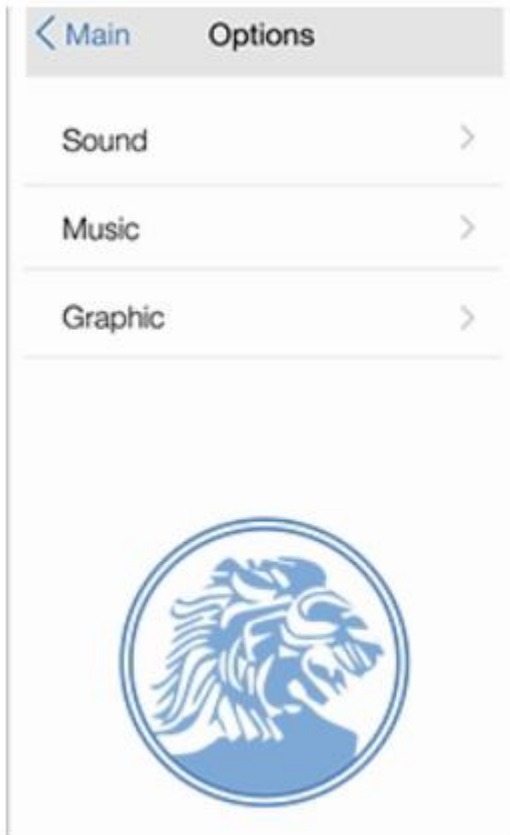
## 5) Navegabilidad

La gente tiende a ignorar la experiencia de navegación en una aplicación a menos que no cumple con sus expectativas. Su trabajo es poner en práctica la navegación de manera que soporta la estructura y el propósito de su aplicación sin necesidad de llamar la atención sobre sí mismo.



# Diseño de Interfaz Grafica (UI)

## 5) Navegabilidad





# Diseño de Interfaz Grafica (UI)

## 6) Contexto Modal

Es un modo de ejecución de ventanas en el que algo existe o es experimentado-tiene ventajas y desventajas. Se puede dar a los usuarios una manera de completar una tarea o para obtener información sin distracciones, pero lo hace impidiendo temporalmente de la interacción con el resto de la aplicación.

Considere este contexto para:

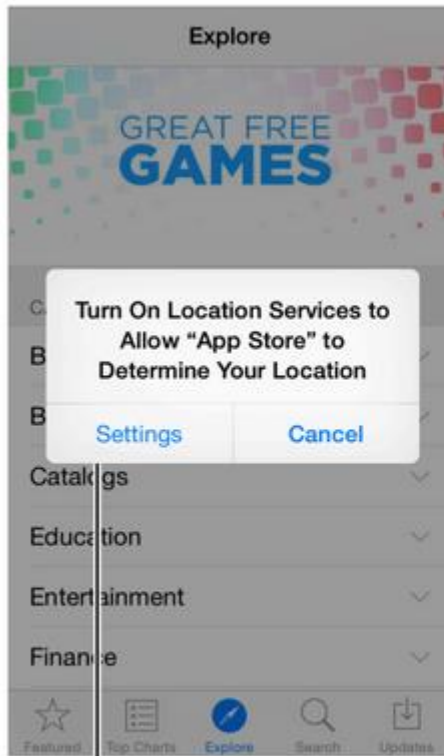
- Cuando el proceso es de vital importancia para conseguir la atención del usuario.
- Es una tarea autónoma y debe ser completada o abandonada explícitamente para evitar dejar los datos del usuario en un estado ambiguo.



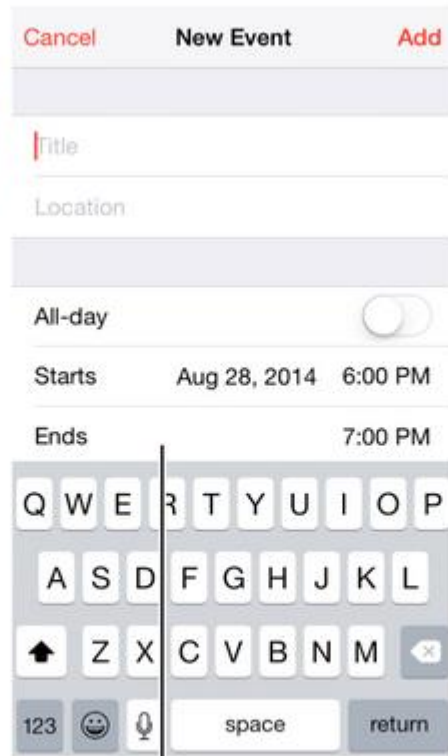


# Diseño de Interfaz Grafica (UI)

## 6) Contexto Modal



Alert



Modal view



Share screen



# Diseño de Interfaz Grafica (UI)

## 7) Interactividad y Comentarios

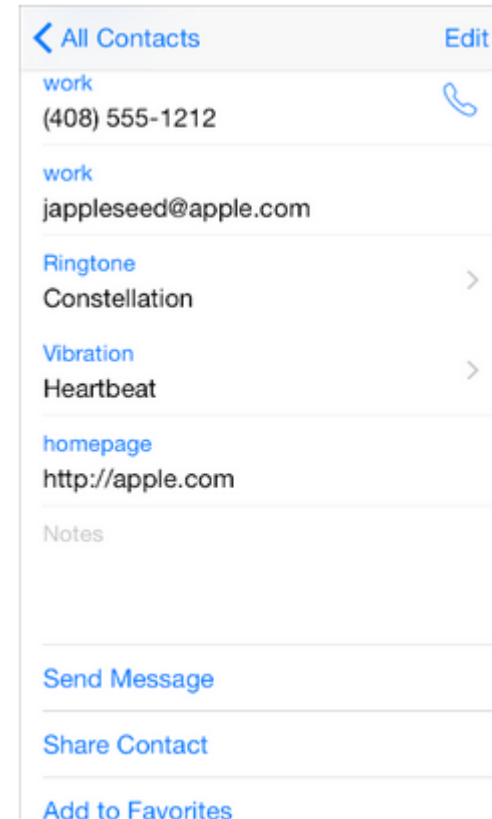
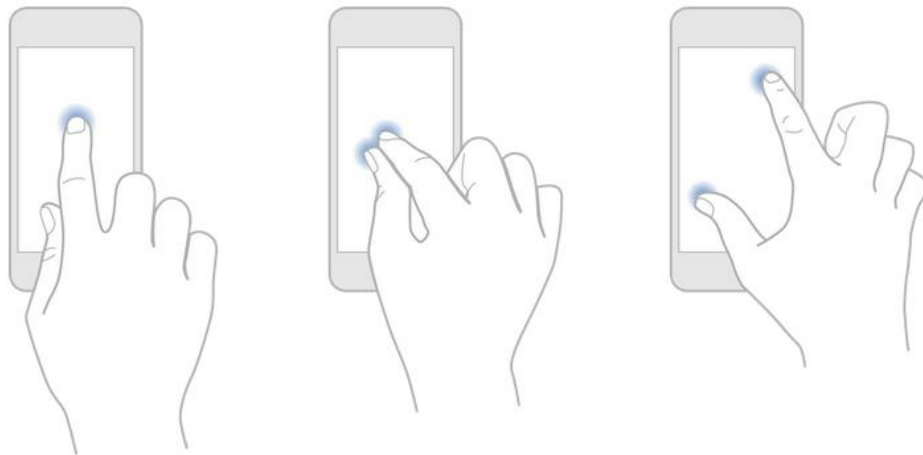
Usuarios conocen los gestos estándar

La gente usa gestos, tales como tocar, arrastrar, y pellizcar para interactuar con aplicaciones y sus dispositivos iOS. Usando gestos da a la gente una conexión personal cerca de sus dispositivos y mejora su sentido de la manipulación directa de objetos que aparecen en pantalla



# Diseño de Interfaz Grafica (UI)

## 7) Interactividad y Comentarios

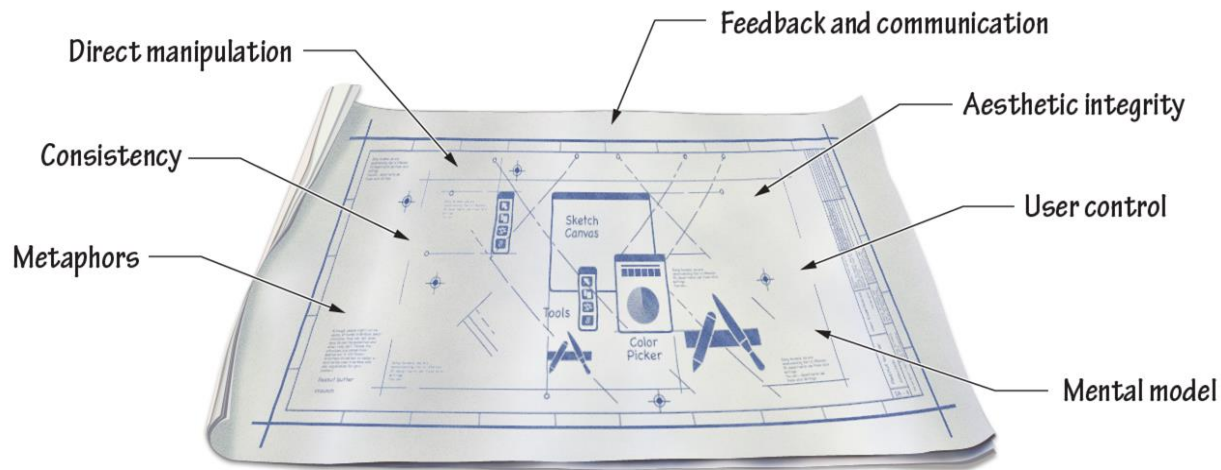




# Diseño de Interfaz Grafica (UI)

## Otros elementos a considerar...

No es suficiente saber crear una aplicación que funcione. Los usuarios esperan aplicaciones de iOS para ser intuitivo, interactivo y atractivo. Piensa en la experiencia del usuario al diseñar todos los aspectos de su aplicación, a partir de las características que elija la forma en que su aplicación responde a un gesto.



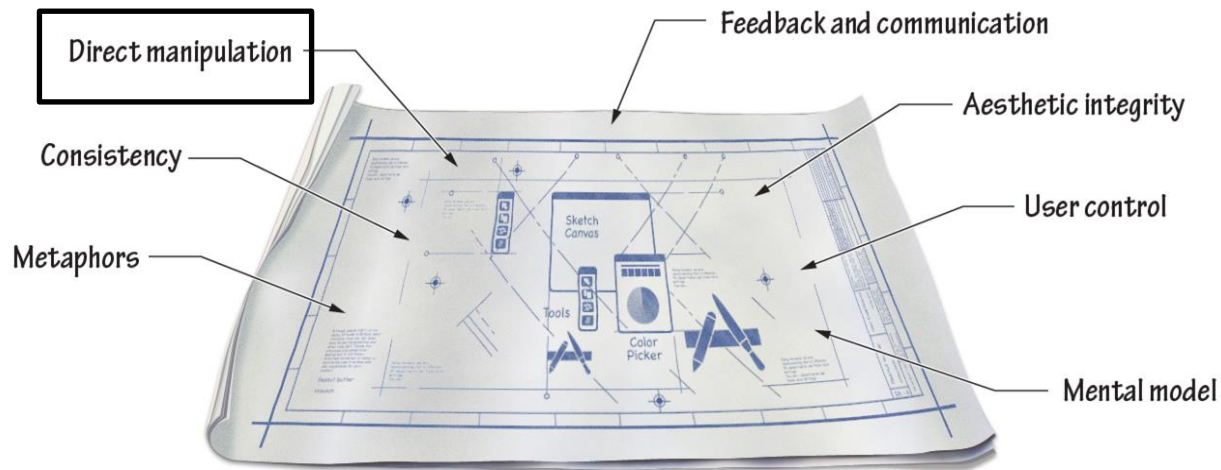


# Diseño de Interfaz Grafica (UI)

Otros elementos a considerar...

## La manipulación directa.

Cuando la gente puede manipular directamente los objetos que aparecen en pantalla en lugar de utilizar controles separados para manipularlos, son más comprometidos con la tarea y comprender más fácilmente los resultados de sus acciones.



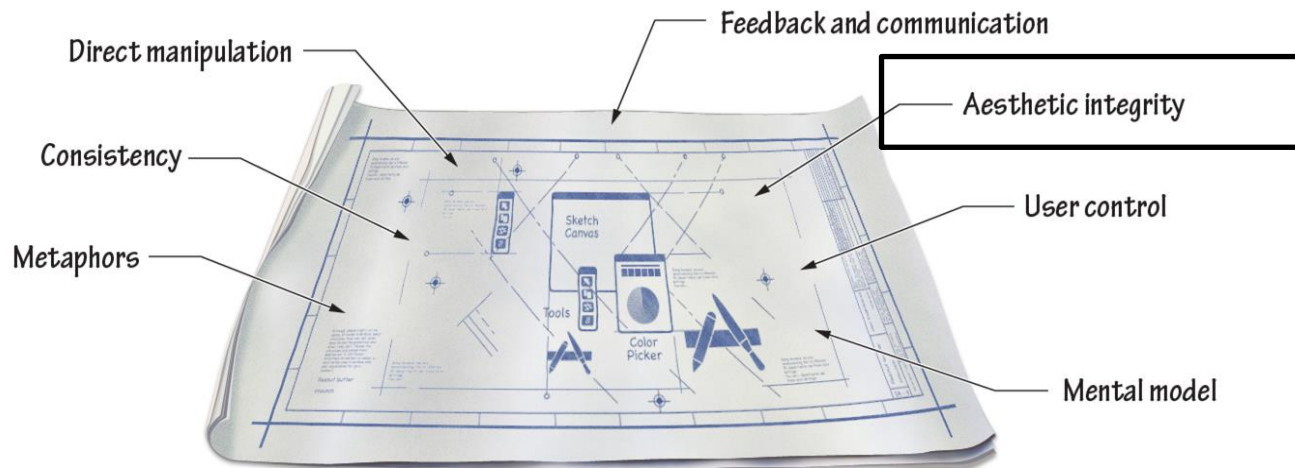


# Diseño de Interfaz Grafica (UI)

Otros elementos a considerar...

## Integridad estética.

Integridad estética no es una medida de cuán bonita es una aplicación, sino que es una medida de qué tanto aplicación se integra con su función.



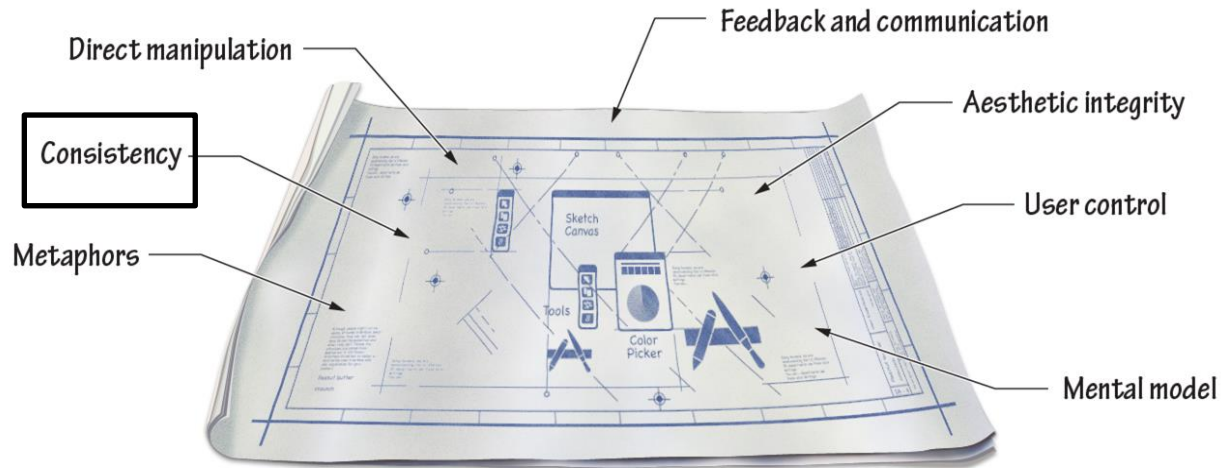


# Diseño de Interfaz Grafica (UI)

## Otros elementos a considerar...

### La consistencia.

La consistencia en la interfaz permite a las personas para transferir sus conocimientos y habilidades de una aplicación a otra. Lo ideal sería que una aplicación es compatible con los estándares de iOS, dentro de sí, y con las versiones anteriores de sí mismo.



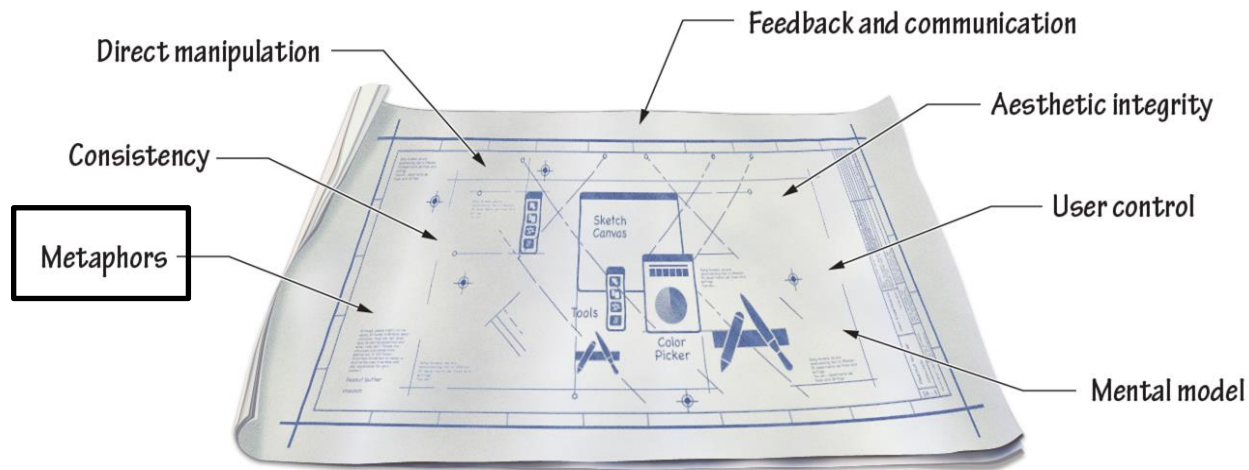


# Diseño de Interfaz Grafica (UI)

Otros elementos a considerar...

## Metaforas

**Metáforas.** Cuando los objetos virtuales y acciones en una aplicación son metáforas de objetos y acciones en el mundo real, los usuarios deben entender cómo usar la aplicación. Las metáforas más adecuadas sugieren un uso o una experiencia sin cumplir las limitaciones del objeto del mundo real o una acción en la que está basada.





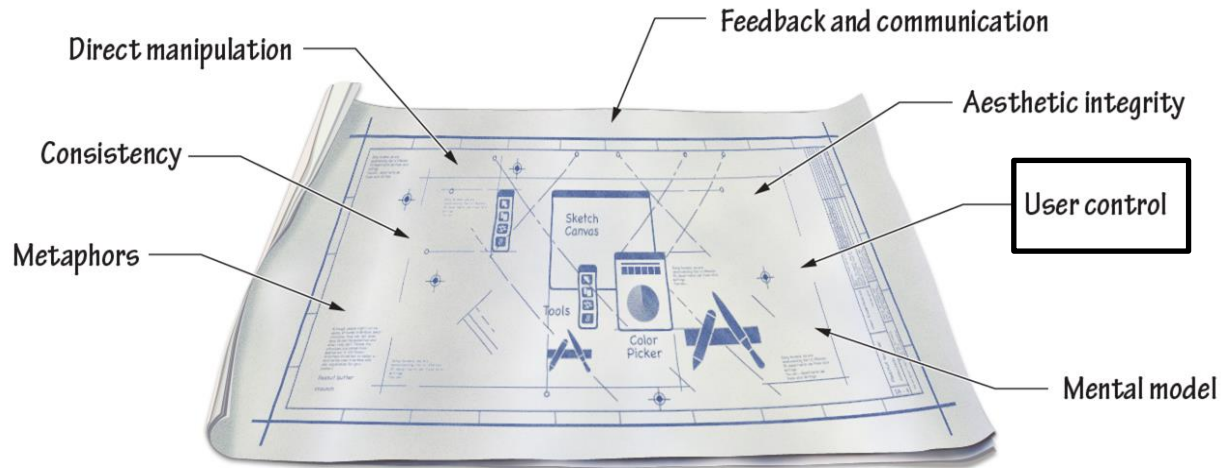


# Diseño de Interfaz Grafica (UI)

## Otros elementos a considerar...

### Control del Usuario.

Aunque una aplicación puede sugerir un curso de acción o advertir sobre las consecuencias peligrosas, por lo general es un error que la aplicación tomar decisiones lejos del usuario. Las mejores aplicaciones para encontrar el equilibrio correcto entre dar a las personas las capacidades que necesitan y ayudarles a evitar resultados peligrosos.





# Lenguaje de desarrollo para IOS

## Extensiones

Extensión	Descripción
.h	Archivos de cabecera. Archivos de cabecera contienen tipo de clase, función, y las declaraciones constantes.
.m	Archivos de implementación. Un archivo con esta extensión puede contener tanto Objective-C y el código C. A veces se llama un archivo de fuente.
.mm	Archivos de implementación. Un archivo de aplicación con esta extensión puede contener código C ++, además de Objective-C y el código C. Use esta extensión sólo si en realidad se refieren a clases C ++ o características de su código de Objective-C.



# Lenguaje de desarrollo para iOS

Objective-C es un lenguaje orientado a objetos elegante que los poderes de todas las aplicaciones de iOS. Usted escribe código Objective-C para crear su aplicación, y que hay que entender este lenguaje con el fin de utilizar la mayoría de los entornos de aplicaciones. Aunque se pueden utilizar otros idiomas para el desarrollo, no se puede construir una aplicación para iOS sin Objective-C.

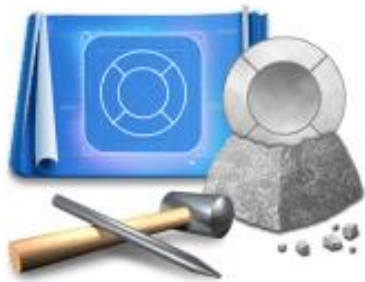




# Lenguaje de desarrollo para IOS

## Tareas Básicas

Después de que ha aprendido lo suficiente acerca de Objective-C para leer y escribir código básico, se puede comenzar a pensar en términos de los objetos. Así como podríamos pensar acerca de los objetos del mundo real, es necesario pensar acerca de lo que un objeto contiene, lo que se puede hacer con él, y cómo relacionarlos con otros objetos.



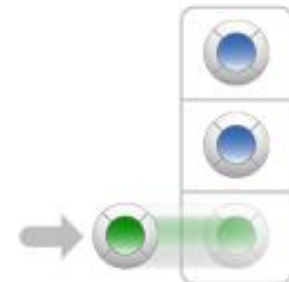
Create



Compare



Query



Access



# Lenguaje de desarrollo para IOS

## Frameworks

Una aplicación(app) consiste en un código que el programador escribe en los frameworks proporcionados por Apple. Un framework contiene una biblioteca de métodos que su aplicación puede llamar. Más de una aplicación puede tener acceso a una biblioteca del framework al mismo tiempo





# Lineas Guia para un buen diseño de interfaz humana

- Centrarse en la tarea principal.
- Hacer uso fácil y obvio.
- Usar terminología centrada en el usuario.  
Haga objetos del Tamaño de la de la yema de los dedos.  
Des -enfaticar en la configuración.
- Utilice la interfaz de usuario (UI) de forma coherente.
- Utilice animación sutil para comunicarse.
- Pregunte a las personas sobre guardar sólo cuando sea necesario.



# Antes de iniciar el desarrollo de Apps para IOS

- Antes de comenzar a escribir una línea de código, debe tomar algunas decisiones críticas de proyecto. Sea lo más específico posible sobre el propósito de su aplicación y características. Elija el tipo de modelo de datos que su aplicación va a utilizar. Decidir sobre un estilo de interfaz de usuario de su aplicación, por ejemplo, en caso de seguir un patrón maestro-detalle o la de una aplicación de utilidades? ¿Quiere que su aplicación sea universal, es decir, una aplicación que se ejecuta en el iPad, iPhone e iPod touch? Estas y otras decisiones de diseño le ayudará a construir la arquitectura de su aplicación.

# Demo Xcode

Aplicación Básica de 3 operaciones (suma, resta y multiplicación) .

Ver Guía de laboratorio 01\_b