

# Programación en Móviles Avanzado

## LABORATORIO 04

### Entorno de Desarrollo XCode y Ciclo de Vida de una Aplicación

#### Programación en Móviles Avanzado

## LABORATORIO 04

### Entorno de Desarrollo XCode y Ciclo de Vida de una Aplicación

CODIGO DEL CURSO:

<b>Alumno(s):</b>				<b>Nota</b>		
<b>Grupo:</b>					<b>Ciclo: V</b>	
<b>Criterio de Evaluación</b>	<b>Excelente (4pts)</b>	<b>Bueno (3pts)</b>	<b>Regular (2pts)</b>	<b>Requiere mejora (1pts)</b>	<b>No acept. (0pts)</b>	<b>Puntaje Logrado</b>
Reconoce la estructura de la interfaz de Xcode y ejecuta aplicativo en emulador y dispositivo real						3
Entiende el proceso de ciclo de vida de un viewController y de la aplicación						3
Desarrolla adecuadamente los ejercicios propuestos						6

Realiza observaciones y conclusiones que aporten un opinión crítica y técnica						3
Es puntual y redacta el informe adecuadamente sin copias de otros autores						2
Evidencia avance en laboratorio						3

#### I.- OBJETIVOS:

- Desarrollar Aplicaciones usando el entorno Xcode
- Conocer los estados manejados por el ciclo de vida de una aplicación

#### II.- SEGURIDAD:

	<b>Advertencia:</b> <b>En este laboratorio está prohibida la manipulación del hardware, conexiones eléctricas o de red; así como la ingestión de alimentos o bebidas.</b>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

#### III.- FUNDAMENTO TEÓRICO:

Revise sus diapositivas del tema antes del desarrollo del laboratorio.

#### IV.- NORMAS EMPLEADAS:

No aplica

#### V.- RECURSOS:

- En este laboratorio cada alumno trabajará con un equipo con MAC OS.

#### VI.- METODOLOGÍA PARA EL DESARROLLO DE LA TAREA:

- El desarrollo del laboratorio es individual.

#### VII.- PROCEDIMIENTO:

### CREACION DE PROYECTOS EN XCODE

#### • XCODE

**Xcode** es el entorno de desarrollo integrado (IDE, en sus siglas en inglés) de Apple Inc. y se suministra gratuitamente junto con Mac OS X. **Xcode** trabaja conjuntamente con **Interface Builder**, una herencia de **NeXT**, una herramienta gráfica para la creación de interfaces de usuario.

- Abra Xcode
  - En la parte derecha, veremos los proyectos recientemente creados.
  - A la izquierda, como parte inicial del curso, veremos la opción de **Create a new Xcode Project**, para crear proyectos nuevos para iphone, ipad e incluso MAC.

- La opción **Clone an existing Project**, sirve para colaborar en un proyecto más de una persona, algo similar a GIT.
- Clic en **Create a new Xcode Project**.  
  
En la pantalla, aparecerán los **templates** disponibles para crear aplicaciones. Que son bases para aplicaciones como de tipo Juego, Master-Detail, para aplicaciones grandes como el calendario en un ipad por ejemplo.  
En las pestañas podemos observar que tenemos diferentes categorías como iOS, WatchOS, etc..
- Clic en **Single View App** o **App**(dependera de la version con la cual trabaje), Clic en **Next**.
- Llene los campos como se indica a continuación:
  - **Product Name:** miPrimerApp
  - **Team:** No haga cambios (sirve para cuando publiquemos nuestro programa en la AppStore o para emular la App en un dispositivo real).
  - **Organization Name:** empresa
  - **Organization Identifier:** com.miempresa
  - **Bundle Identifier:** miempresa.com.miPrimerApp (se llenará por defecto).
  - **User Interface:** Storyboard
  - **Language:** Swift
  - Quite los checks debajo de **Language**.
  - Clic en **Next**.
- En la siguiente pantalla, escoja un directorio para crear el proyecto.
  - Escoja **Documents**
  - Cree una subcarpeta de nombre "**Proyectos iOS**".
  - Dentro de la última carpeta creada una sub carpeta de nombre: "**EntornoXcode**".
  - Si desea puede dejar habilitado el Check de "**Create Git repository on**"
  - Clic en **Create**.

**Navigation Area:** Área en la que podremos ver los archivos creados de manera automática en nuestro proyecto. Los archivos variarán de acuerdo al tipo de template que elijamos a la hora de la configuración.

**Code Area:** Área en la que codificaremos. En esta pantalla inicial aparecen algunas opciones finales de configuración como el nombre del paquete, la versión del proyecto, la orientación en la que la aplicación se ejecutará, etc.

**Utility Area:** Área en la que haremos uso de propiedades de objetos que se vayan insertando a Interface Builder.

- Clic en el archivo **ViewController.swift**. Al lado derecho se mostrar el código del archivo
- Clic en **Main.storyboard**: Este archivo abrirá una pre-visualización de la vista del viewController creado por defecto.

- En la parte superior busque el botón de **Library** .
- Este botón permite mostrar la librería de objetos que se pueden utilizar para crear aplicaciones en iOS. Haga clic sobre este botón, se mostrar la lista de objetos de la librería
- En la ventana flotante mostrada, escriba **LABEL**.
- Arrastre el elemento al medio de la pantalla.
- Doble clic en el **Label** creado y modifique el texto por: **“Hola Mundo”**.
- Clic en el **label** creado.
- Clic en el botón **“Show the attributes Inspector”**, ubicado en el **Utility Area**. Esta pestaña que acabamos de activar, sirve para ver las **propiedades** del Label, tales **como texto, color, tamaño de fuente, etc.**
- Pruebe cambiar la propiedad **Color** y **Font** a su preferencia

### Jump Bar

- Seleccione nuevamente la etiqueta creada.
- Observe detenidamente la **JUMP BAR**.

### ¿Qué es y para que sirve la Jump Bar en MAC ?

## EJECUTAR MI PRIMERA APLICACIÓN

### • EMULADOR:

**PASO 1:** Ubique la lista de dispositivos para emular nuestra aplicación en la esquina superior izquierda. Seleccione de la lista el iPhone 11. (Puede elegir cualquiera, pero para este caso tratemos de elegir un emulador que no consuma muchos recursos)

Podremos elegir entre diferentes dispositivos, siempre y cuando en la aplicación

configuremos que el proyecto estará diseñado tanto para iPhone como para iPad, de eso depende la lista de dispositivos.

Sin embargo, si queremos cambiar dichos valores, solo de clic en el nombre del proyecto en el navegador a la izquierda y cambie la configuración.

**PASO 2:** Una vez elegido el dispositivo, Clic en el botón en forma de **Play** o en la **barra de Menú -> Product -> Run**.

**Ojo:** Si se le pregunta si desea **activar el modo desarrollador** haga clic en **enable**

Empezará a ejecutarse el emulador para el dispositivo seleccionado, en el cual se lanzará la aplicación.

- Agregue un **Label** más a su proyecto, en el que se muestre su **Nombre**, con un color de fondo y un tamaño más grande al de la etiqueta previa
- Ejecute su proyecto y adjunte una captura del resultado final



- **DISPOSITIVO REAL (Opcional)**

Primero, conecte el dispositivo móvil (iPad, iPhone, iPod, etc.) al ordenador. Asegúrese de que VMWare lo tenga conectado a la máquina virtual en caso estén usando una máquina virtual.

**OJO:** Si se le pide que de permisos para acceder al dispositivo , haga clic en **Permitir en el dispositivo iPhone, iPad**.

**PASO 1:** En su dispositivo iPhone o iPad haga clic en **Confiar**. Haga que Xcode reconozca el dispositivo conectado. En la esquina superior Izquierda, escoja el dispositivo conectado. Debería aparecer una opción como: **“iPad de [Nombre]”**.

**PASO 2:** Clic en el botón de **play** o en la barra de **Menú ->Product->Run**

- Se generará un error ya que no se ha registrado aun la cuenta de desarrollador creada en el anterior laboratorio. Haga clic en la esquina superior derecha(icono rojo) para visualizar el error
- Se mostrará el siguiente mensaje

- En **Navigation Area** haga clic en el símbolo de la **Carpeta** y luego haga clic en el nombre de su proyecto **miPrimerApp**
- En la pestaña de **General**, vaya al grupo **Signing** y haga clic en **Add Account ...**
- Proporcione su ID y contraseña creada en Apple
- Una vez agregada la cuenta cierre la ventana
- Diríjase nuevamente al grupo **Signing** y en el campo de **Team** elija su cuenta agregada
- Espere a que se descarguen los certificados para uso de la aplicación
- **OPCIONAL:** (Solo para versiones antiguas de xcode) Diríjase a la pestaña de **Capabilities** y active la opción de **App Groups** y posteriormente **Keychain Sharing**. (Espere a que en ambos casos valide las credenciales de su ID)
- Vuelva a la pestaña de **General** y verifique que en el grupo **Signing** su obtuvieron las credenciales para la aplicación
- Ahora si puede ejecutar su proyecto. Asegúrese que esta seleccionado en la lista de emuladores su dispositivo físico y haga clic en el botón de **Play**. Si le solicita clave de llavero, ingrese su contraseña de su usuario super.

**ADICIONAL:** Si no reconoce, realizar los pasos listados abajo.

- Se intentará instalar la aplicación en el dispositivo iPhone o iPad pero votara un error
- Este error se produce porque hay que conceder permisos en el dispositivo físico para que pueda instalar la aplicación
- Diríjase al iPad o iPhone y vaya a **ConfiguracionGeneralAdmon. De dispositivos**
  - Aquí se mostrará su ID de Apple, haga clic sobre este y se mostrará la aplicación que estamos intentando instalar. Haga clic en **Confiar en "su ID de Apple"**. Confirme una vez más para verificar el proceso
- Ahora sí, vuelva a ejecutar el proyecto. Debería de mostrarse en el dispositivo sin ningún inconveniente, pero si por alguna razón obtiene el siguiente error de la figura, modifique como se indica en el paso siguiente
- Dirijase a menu **ProductSchemeEdit Scheme...**
  - Deshabilite el check de **Debug Executable**
- Vuelva a ejecutar el proyecto y verifique que se instala en el dispositivo
- OJO: Sino reconoce el App intente instalar el certificado que se indica

<https://www.apple.com/certificateauthority/AppleWWDRCA3.cer>

- **Adjunte captura de la ejecución del proyecto en el dispositivo real**

## FUENTE

Para ver el número de las líneas de código y personalizar la fuente en la que vamos a desarrollar:

- Clic en el Menú **Xcode -> Preferences**.
- Abra la pestaña **"Text Editing"**
- Active el Check de **"Line numbers"**.
- Ahora ubíquese en la pestaña **"Fonts & Colors"**
  
- Pruebe modificando algunas características.
- Adjunte capturas de las características probadas



## CICLO DE VIDA DE UNA APLICACIÓN

- Abra el archivo **ViewController.swift**
- Sobre escriba las funciones nativas de la clase **UIViewController** para que el código quede de la siguiente forma:
  
- Abra el archivo **SceneDelegate.swift** y sobre escriba las funciones delegadas de la siguiente manera:
  
- Abra el archivo **Main.storyboard**. Haga clic en la librería de objetos y digite la palabra **button** e inserte el botón debajo del **Label "Hola Mundo"**. Coloque como nombre del botón **"Ir al segundo viewController"**. Haga clic en la librería de objetos y digite la palabra **viewController** y arrastre este a la pantalla de **Interface Builder**. Debe quedar como se muestra
  
- Inserte un **Button** en el nuevo **viewController** insertado que tenga como nombre **"Ir a primer viewController"**
  
- Se va a proceder a llamar al **segundo viewController** cuando se haga clic en el **Boton "Ir a segundo viewController"**
- Haga clic en **Boton "Ir a segundo viewController"** y mantenga presionada la tecla de **Ctrl**, sin soltar lleve el puntero hasta el **nuevo viewController** insertado y suelte el clic del mouse. En la ventana mostrada elija **Show**. Se creará una flecha entre el **primer viewController** y el **segundo viewController** para indicar la dirección de la llamada a la vista
  
- Repita el mismo procedimiento para que el botón **"Ir a primer viewController"** llame al primer **viewController**. Debe quedar como se muestra

- Vaya al menú **FileNewNew File**
  - Elija **Cocoa Touch Class** y **Next**
    - Class: **ViewController2**
    - Subclass of: **UIViewController**
    - No active la casilla de **Also create XIB file**
    - Language: **Swift**
    - Clic en **Next** y luego **Create**
- Copie las mismas funciones creadas en **viewController.swift** a **viewController2.swift**, solo que haga referencia a que se esta en **viewController2**. El archivo **viewController2.swift** debe quedar como se muestra
- Haga clic **Main.storyboard**, luego haga clic en el icono amarillo del **segundo viewController** agregado. Después en **Utility Area** haga clic en **Show the Identity Inspector**
- En **Class** busque el archivo **ViewController2.swift**
- Ejecute la aplicación en un simulador o dispositivo físico.
- Notara que cuando se llama de una vista a otra, se apilan a modo de pestañas una encima de otra las vistas creadas. Para evitar esto, podemos cambiar para que al llamar a una vista se muestre en pantalla completa.
- Haga clic en el botón Amarillo del **primer viewController** (como se hizo en el punto 11) después diríjase a **Show the attributes inspector** y en la sección **Presentation** elija **Full Screen**. Repita este mismo proceso para el **segundo viewController**.
- Ejecute nuevamente la aplicación para probar la funcionalidad del ciclo de vida de una aplicación
- En la vista **DEBUG**, indique los cambios que ocurren cuando se abre la aplicación. Explique lo mostrado

- Haga clic en el botón de **“Ir a segundo viewController”**. Comente lo sucedido. Adjunte imágenes

- Haga clic en el botón de **“Ir a primer viewController”**. Comente lo sucedido. Adjunte imágenes



- Presione el botón de **home** (En el menú del emulador elija **HardwareHome**). Comente lo sucedido. Adjunte imagenes

- Regrese a la aplicación del segundo plano a primer plano(En el emulador haga clic en la aplicación creada en su pantalla inicial llamada **miPrimerApp**). Comente lo sucedido. Adjunte imagenes

- Termine la aplicación(En la parte inferior del emulador, se muestra una línea horizontal, deslice para arriba y se mostrar las ventanas abiertas. Elimine de la lista su aplicativo). Comente lo sucedido. Adjunte imagenes

- Indique la diferencia entre **ViewController** y **SceneDelegate** con respecto al control de ciclo de vida de una aplicación

#### TAREA:

- Vea la lista de objetos que proporciona Xcode y cree el siguiente diseño (Solo diseño, no código)

**OBSERVACIONES (5 mínimo):**

(Las observaciones son las notas aclaratorias, objeciones y problemas que se pudo presentar en el desarrollo del laboratorio)

**CONCLUSIONES (5 mínimo):**

(Las conclusiones son una opinión personal sobre tu trabajo, explicar como resolviste las dudas o problemas presentados en el laboratorio. Además de aportar una opinión crítica de lo realizado)