##### GUÍA DE LABORATORIO 06

**“Core Data”**

**I.- OBJETIVOS:**

* Crear una aplicación con persistencia de Datos

**II.- SEGURIDAD:**

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

**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 trabará con un equipo con MAC OS.

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

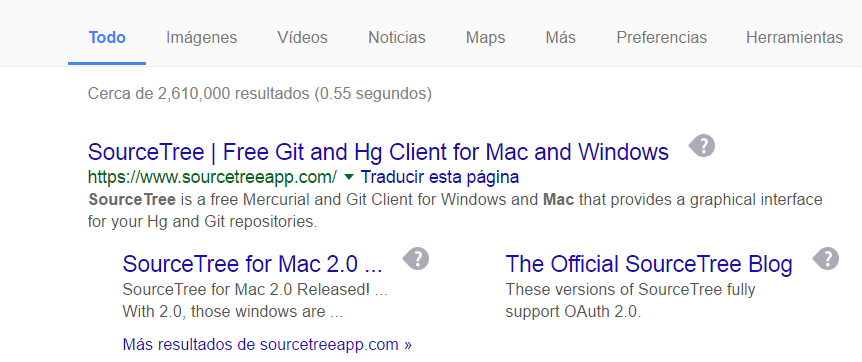
* El desarrollo del laboratorio es individual.

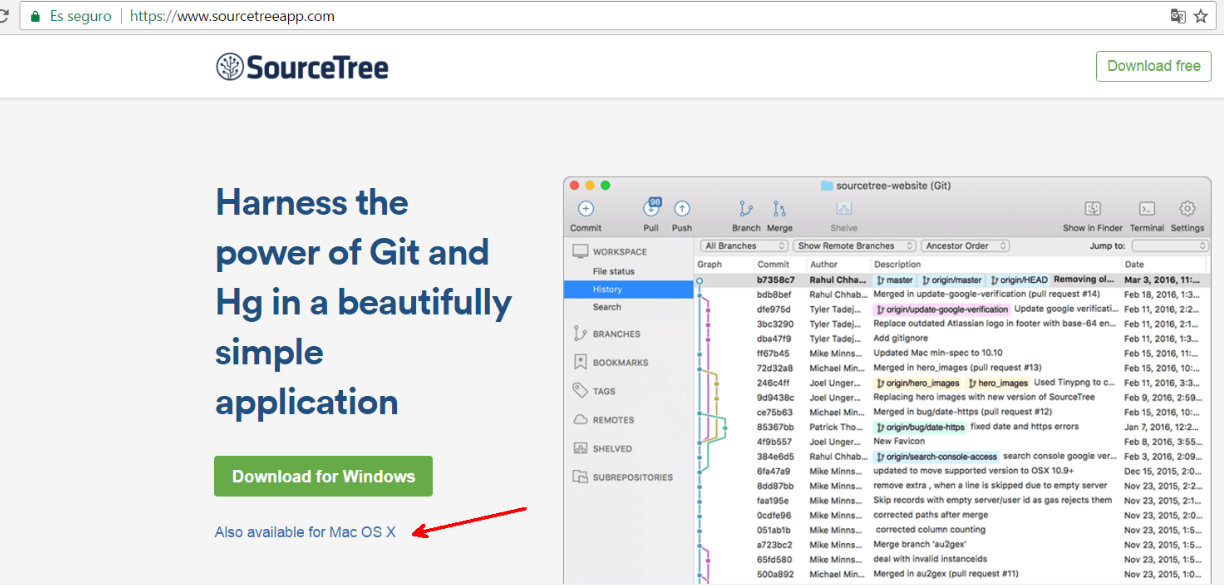
**VII.- PROCEDIMIENTO:**

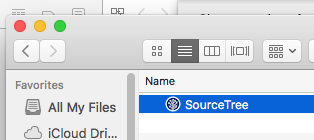
**ACTIVIDADES:**

**INSTALACIÓN DE SOURCETREE**

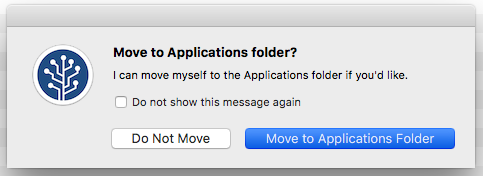
1. Abra Safari y busque la aplicación Source Tree, descargue la versión para MAC



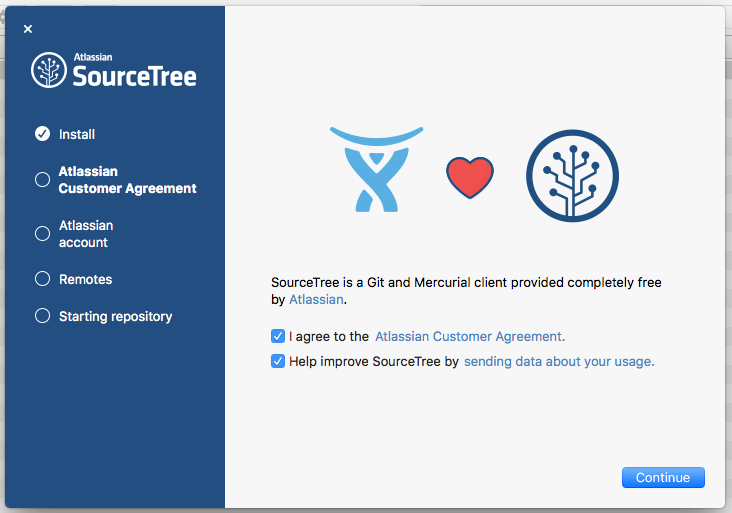




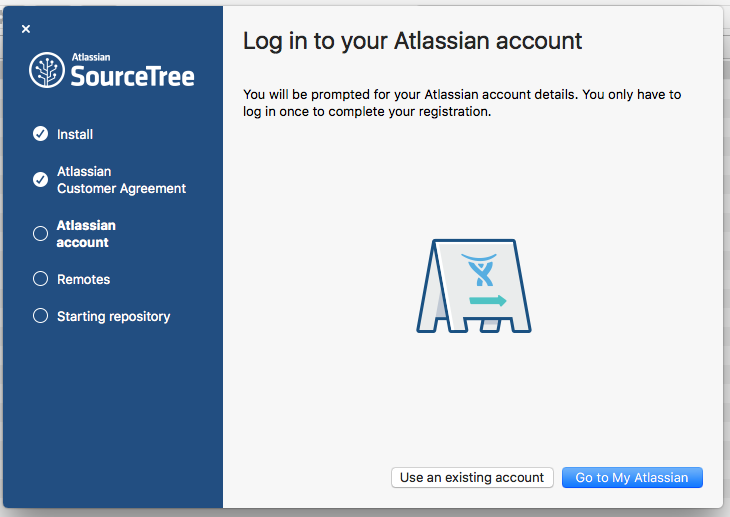
1. Descomprima el archivo e instale en su equipo MAC. Es posible obtener un mensaje de alerta, clic en Open.
2. Luego elija la opción **Move to Applications Folder**



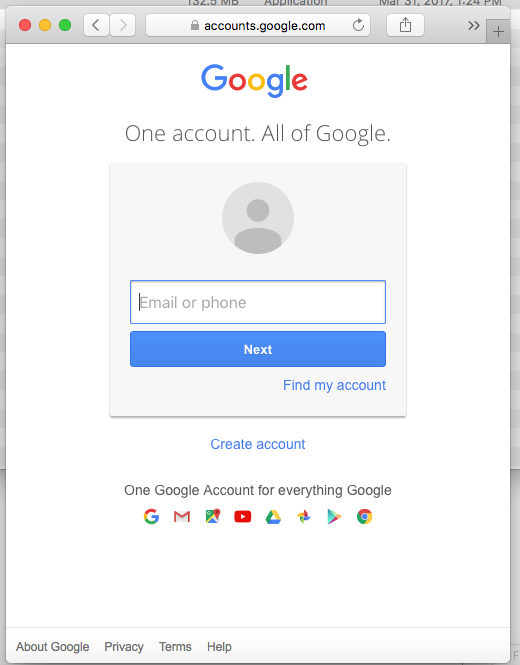
1. Clic en Continue aceptando los términos y condiciones de uso



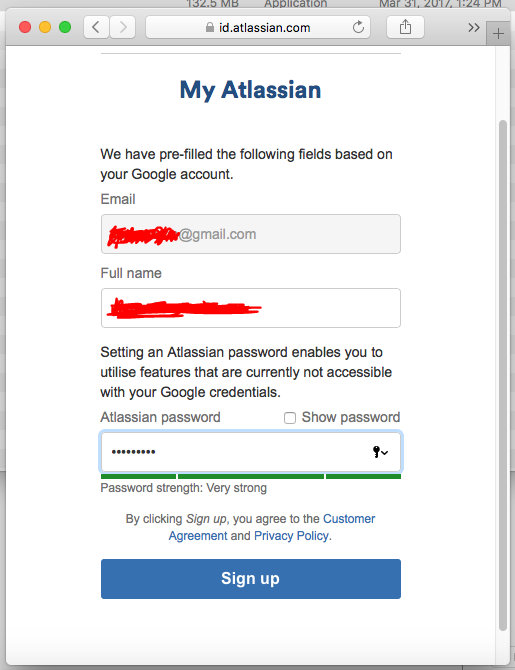
1. Para crear una cuenta de la compañía **Alassian**, clic en **Go to My Atlassian.**



1. Introduzca su correo de Google.



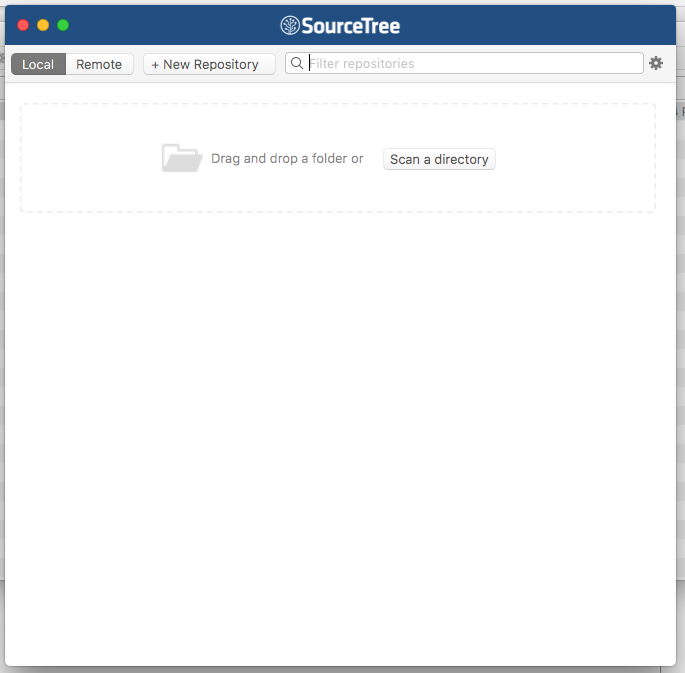
1. Elija permitir y luego, coloque su nueva contraseña.



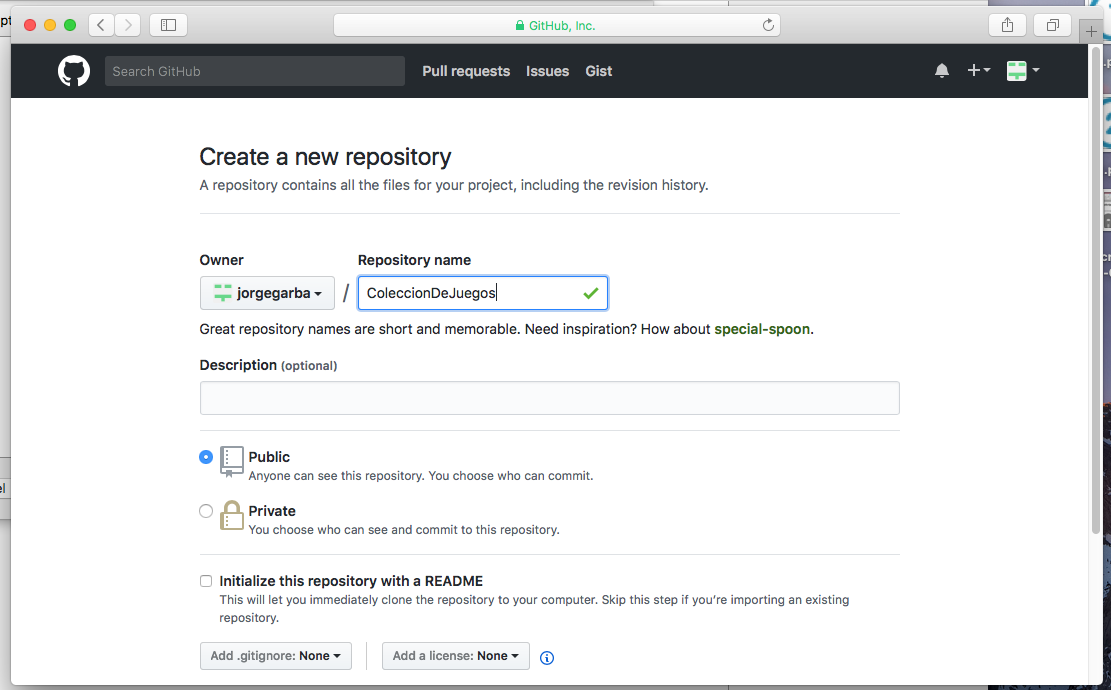
1. Si desea, puede continuar con la configuración de su cuenta. Sin embargo, ya está creada, regrese al instalador de Source Tree y elija **Using an existing account.**
2. Una vez colocado el correo, debería tener el siguiente resultado.



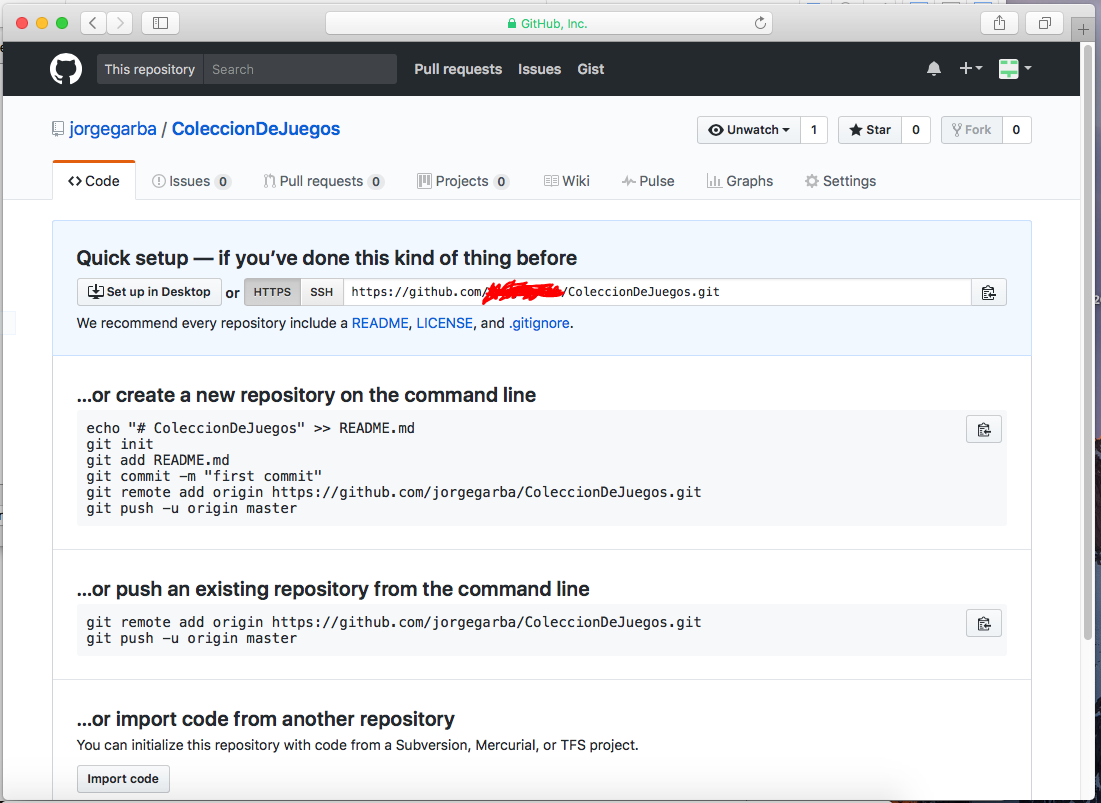
1. En el siguiente menú elija **skip setup.**
2. Finalmente tendrá la interfaz de inicio de Source Tree.



1. Ahora, abra safari e ingrese a su cuenta en **github,** si no tiene una, créela.
2. Cree un nuevo repositorio, asegúrese de colocar el mismo nombre del proyecto que crearemos en esta sesión. “**ColeccionDeJuegos**” (sin tilde)



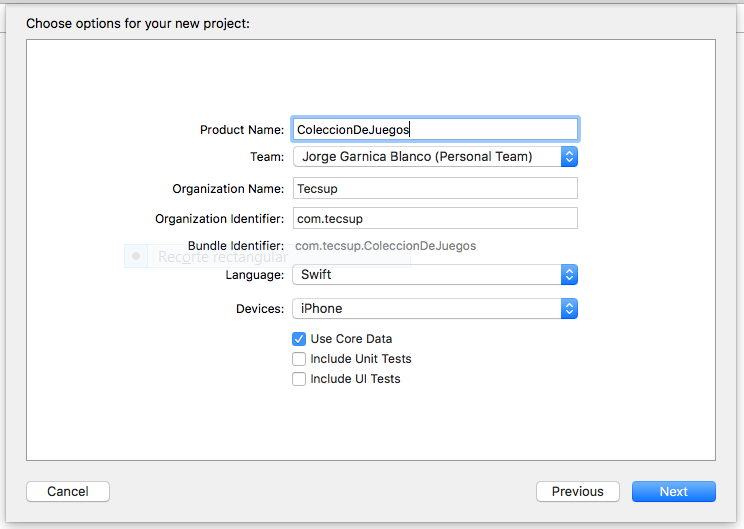
1. Clic en el botón verde de abajo “**Create Repository**”



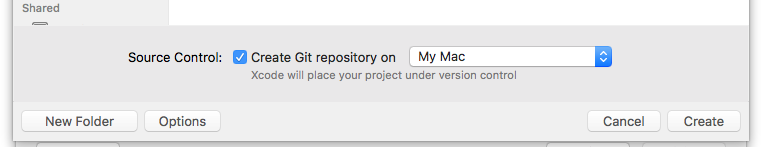
1. Copie la dirección generada del repositorio.

**CREANDO LA APLICACIÓN**

1. Cree un nuevo proyecto en Xcode de tipo Single View Application
2. Complete los siguientes campos



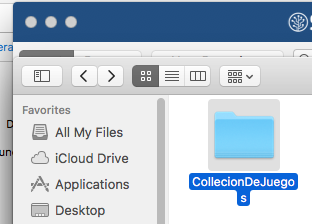
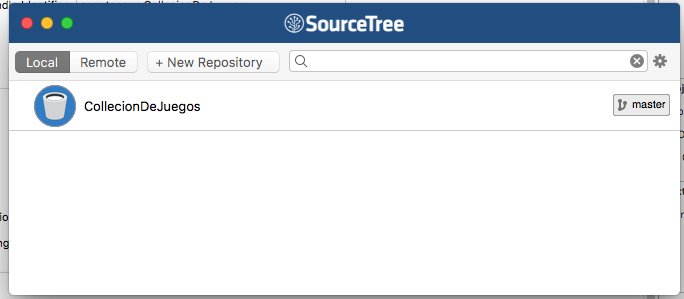
1. En el siguiente menú contextual, elija la opción **Create Git Repository on My Mac.**



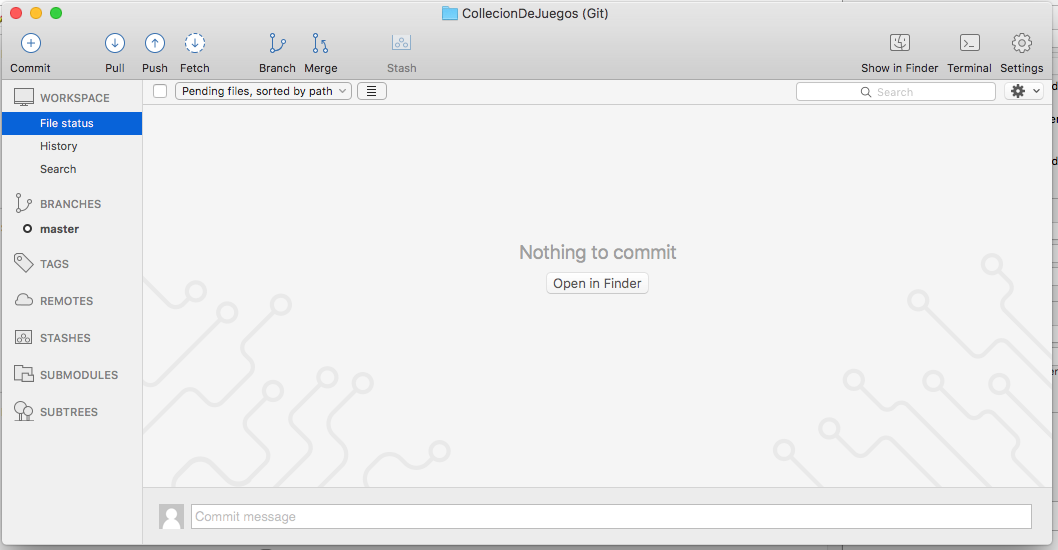
1. Termine la creación del proyecto.

**SINCRONIZANDO EL REPOSITORIO**

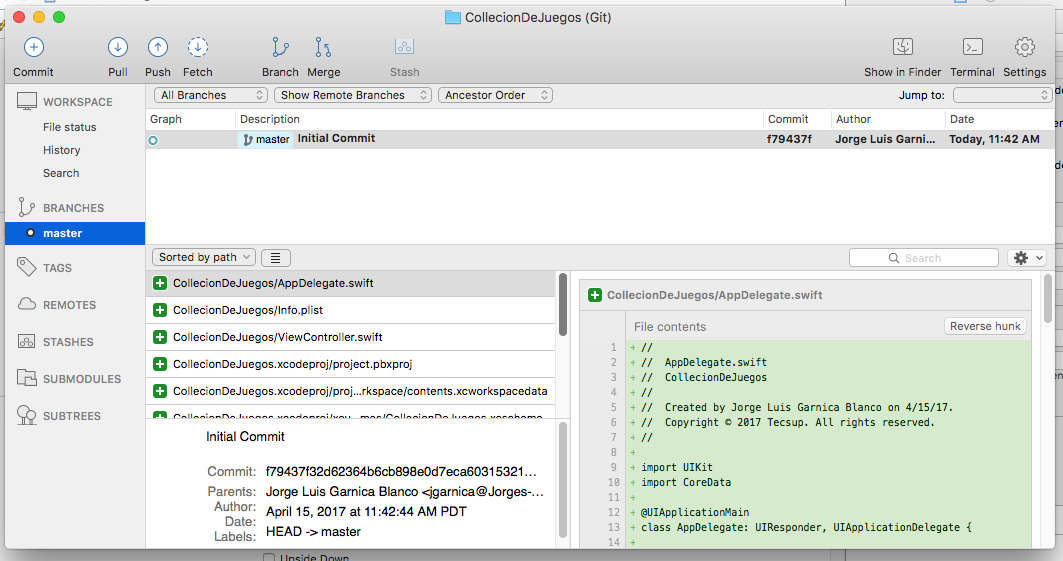
1. Ahora, abra Source Tree.
2. Clic en **New Repository -> Add Existing Local Repository.**
3. Abra la carpeta de creación del proyecto.

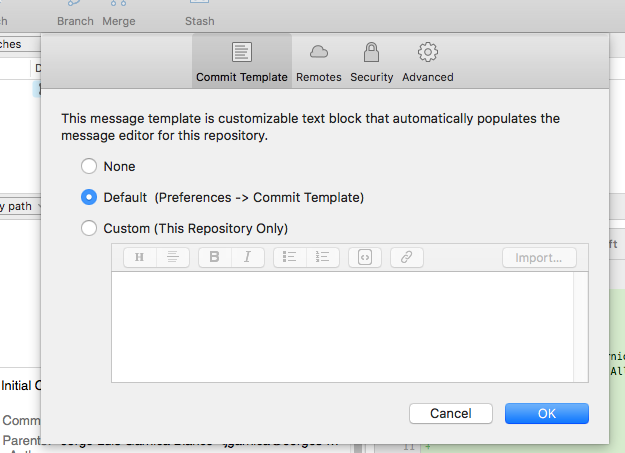
1. Doble clic en el nombre del repositorio y verá que no hay commits realizados aún.



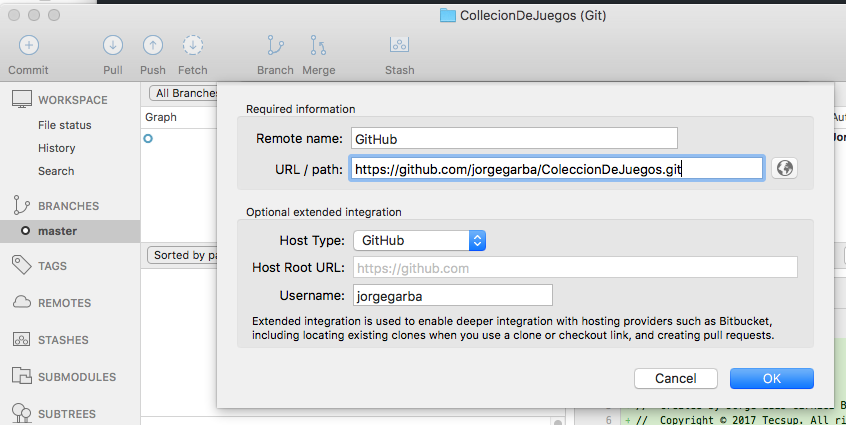
1. Seleccione el menú **master.**



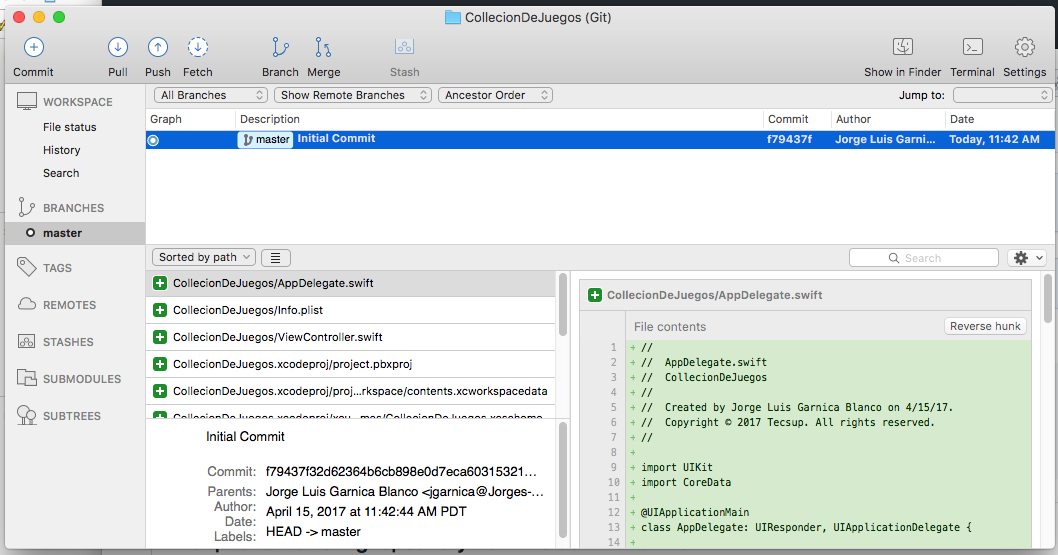
1. Clic en Settings



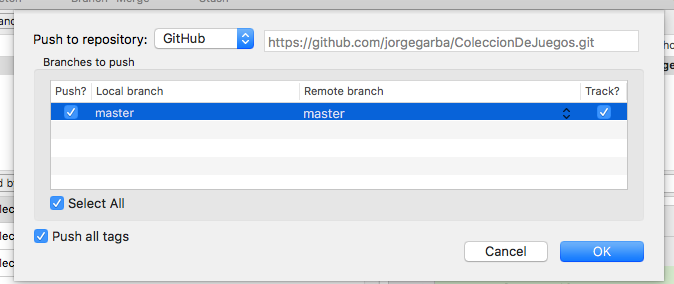
1. Clic en la pestaña **Remotes**, luego clic en **Add**
2. En el nombre coloque GitHub
3. Y en el URL coloque el URL copiado del repositorio creado en GitHub

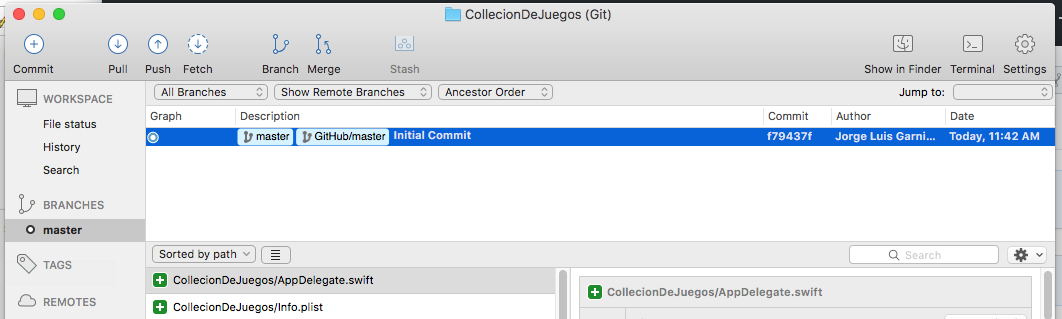


1. Clic en OK y Ok nuevamente.

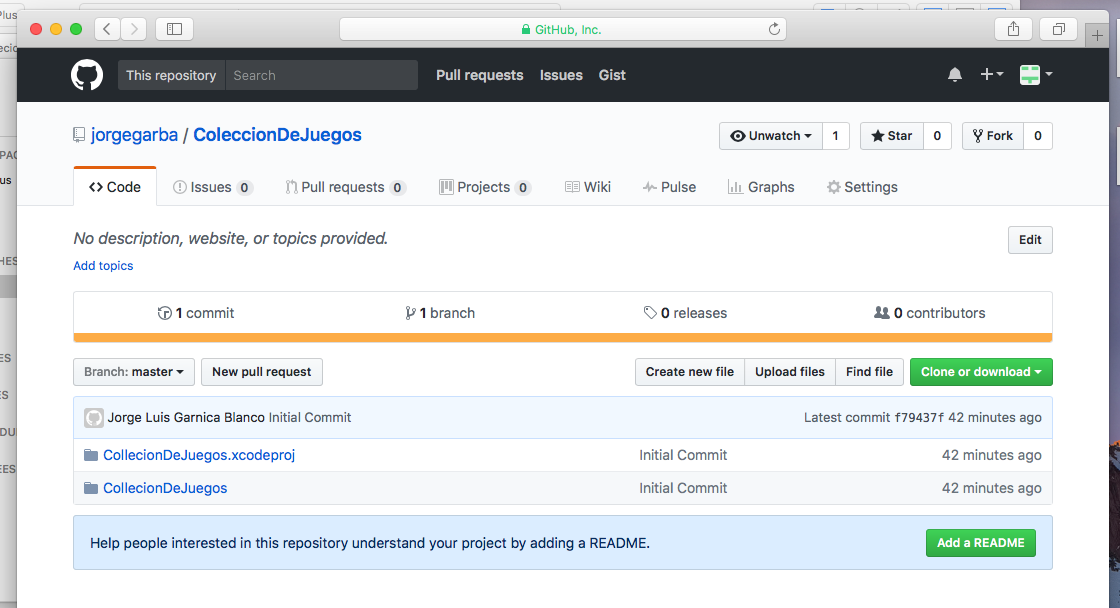


1. Clic en el Botón **Push,** y en el menú contextual escoja la rama master, Luego pulse Ok, si se piden datos de acceso, coloque su cuenta y contraseña de GitHub



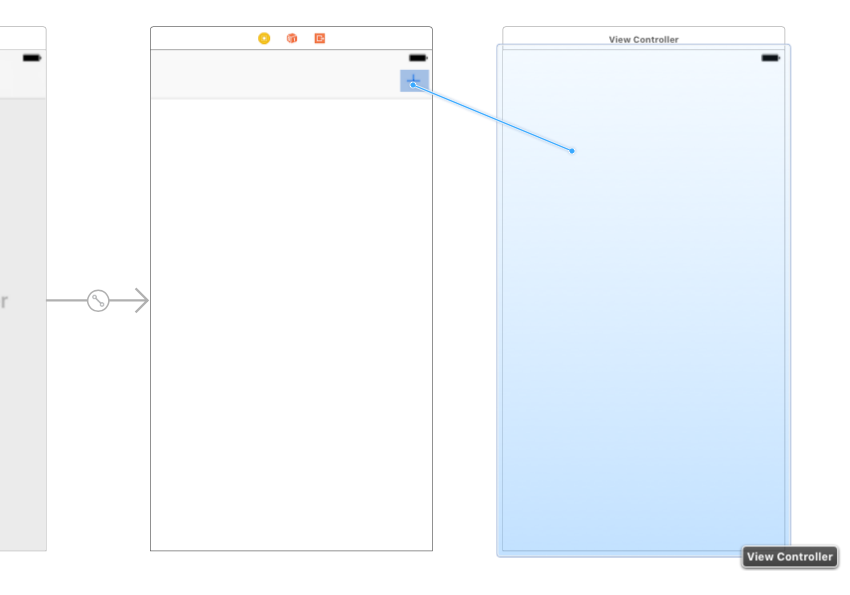


1. Ahora abra safari, ingrese a su repositorio en GitHub, y verifique que los cambios se hayan realizado.

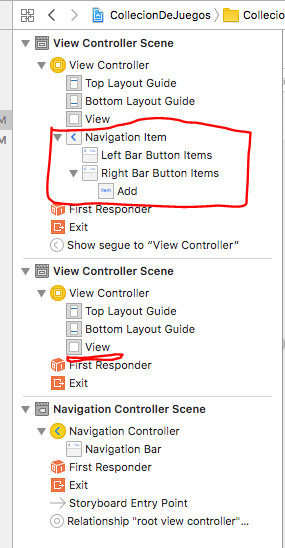


**DISEÑANDO LA INTERFAZ DE USUARIO**

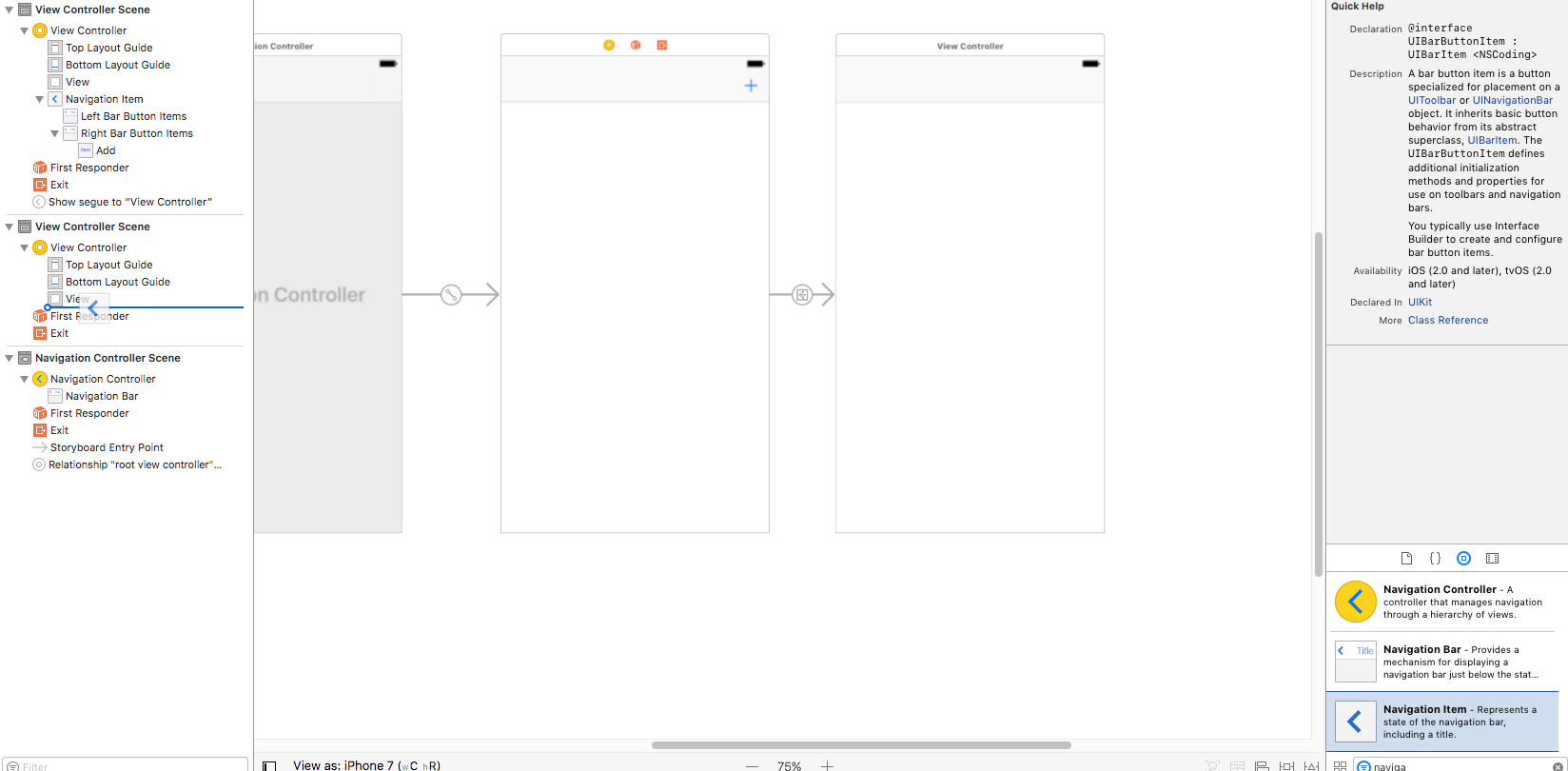
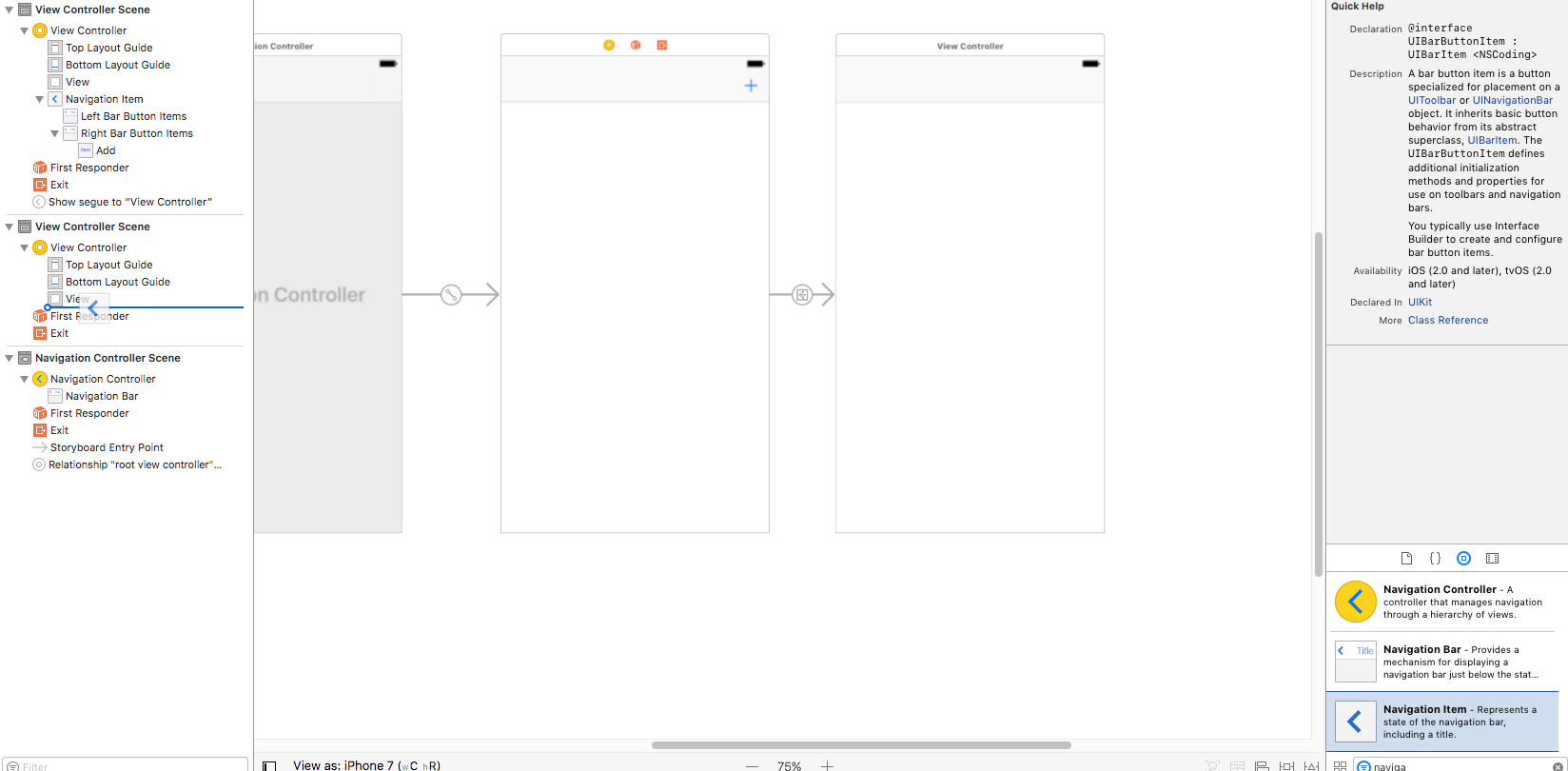
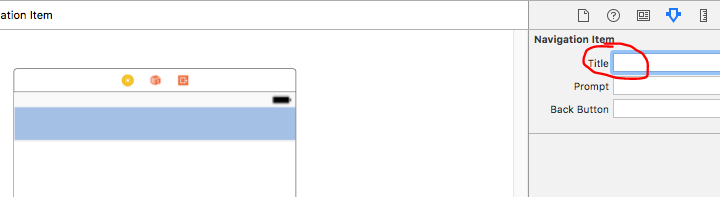
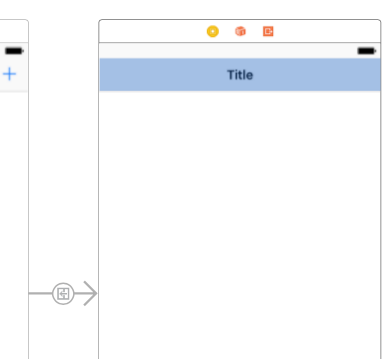
1. En el main.storyboard, cree el Navigation Controller para el ViewController que se crea por defecto.
2. Agregue un elemento Bar Button Item a la esquina superior derecha del ViewController.
3. Seleccione el botón y modifique su propiedad “*System Item*” a “*Add*”
4. Arrastre un segundo ViewController al main.storyboard
5. Cree un nuevo enlace de tipo “Show” del ViewController inicial al recién creado **DESDE EL BOTÓN DE AÑADIR (+)**



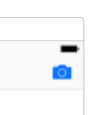
Como podemos observar, en el segundo ViewController no podremos crear botones de navegación en vista de no poseer un **Navigation** **Item**



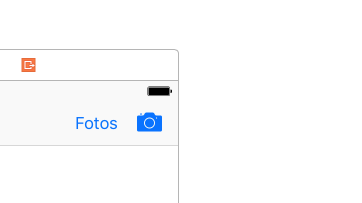
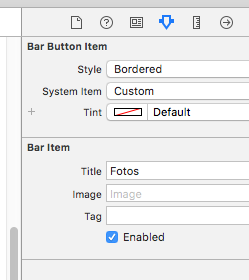
1. En el explorador de objetos, busque un **navigation ítem,** y arrástrelo a la escena del segundo ViewController, Se creará un texto de título por defecto, sólo borre el texto.

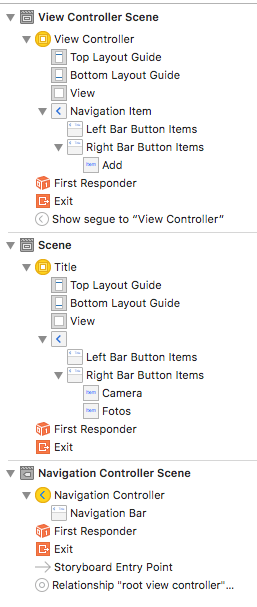
1. Agregue un Bar Button al segundo ViewController.
2. Seleccione el botón y modifique su propiedad “*System Item*” a “*Camera*”



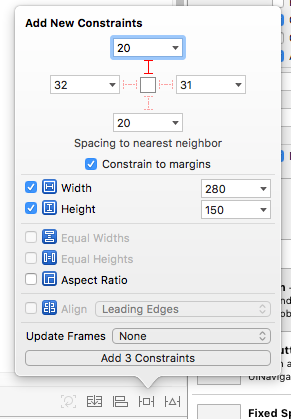
1. Arrastre un segundo Button Bar al costado izquierdo del botón de la cámara y coloque como texto “Fotos”.

1. Las escenas deberían quedar así:



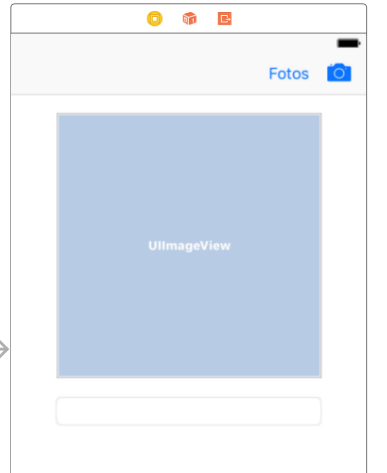
1. En el explorador de objetos, busque un elemento de nombre “ImageView”, y arrástrelo al segundo ViewController.
2. Para el elemento añada los siguientes Constraints:
   1. Horizontalmente Centrado.
   2. Margen Top 20
   3. Marcar Casilla de Width con valor de: 280
   4. Marcar Casilla de Heigth con valor de: 150



1. Actualice los frames:



1. Añada una caja de texto debajo del ImageView con los siguientes constraints:
   1. Centrado horizontalmente.
   2. Espacio vertical dependiente del ImageView con 20.
   3. Cree un nuevo enlace dependiente del ImageView pero elija la opción “Equal Widths” para que tengan el mismo ancho.
   4. Actualice los frames.



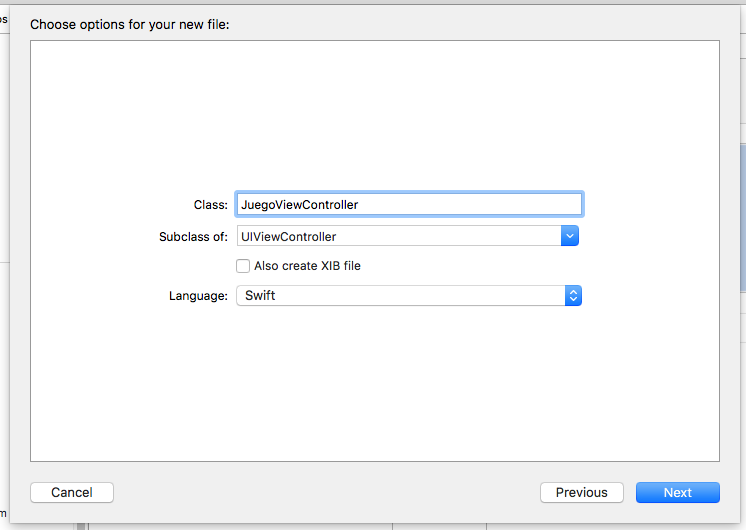
1. Añada un Botón debajo de la caja de texto con los siguientes constraints:
   1. Centrado Horizontalmente.
   2. Espacio vertical de 20 con respecto a la caja de texto.
   3. Cambie el texto a “**Agregar**”
   4. Actualice los frames.



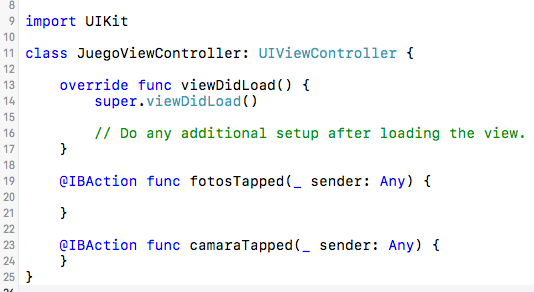
1. Seleccione la caja de texto y en la pestaña de sus propiedades coloque “**Titulo**” en ***placeholder***, y en alineamiento verifique que se encuentre centrado.

**AGREGANDO LOS CONTROLADORES.**

1. Agregue una nueva clase de tipo cocoa touch.



1. Asigne la clase creada como controlador de la vista que tiene el ImageView.
2. Cree un enlace de tipo action del botón **fotos,** a su controlador con el nombre de: fotosTapped (tapped hace referencia haber sido clicado con el dedo).
3. Cree un enlace de tipo action del botón **cámara,** a su controlador con el nombre de: camaraTapped (tapped hace referencia haber sido clicado con el dedo).
4. Adicionalmente, elimine el código que no nos sirve en dicho controler para que quede de la siguiente manera:



1. Cree una conexión de tipo outlet del ImageView hacia su controlador como variable de clase con el nombre de: ***JuegoImageView.***
2. Cree una conexión de tipo outlet del TextField hacia su controlador como variable de clase con el nombre de: ***tituloTextField.***
3. Cree una conexión de tipo action del botón “agregar” hacia su controlador con el nombre de: ***agregarTapped.***

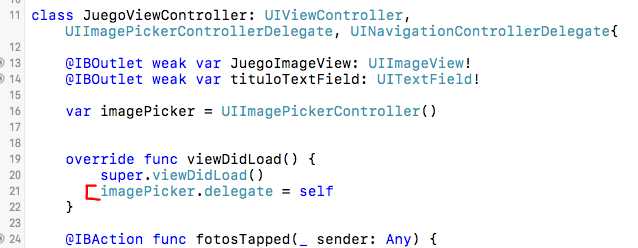
Todo debería quedar de la siguiente forma:



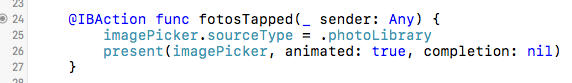
1. Implemente los protocolos para controlar el uso de multimedia en imágenes del dispositivo.



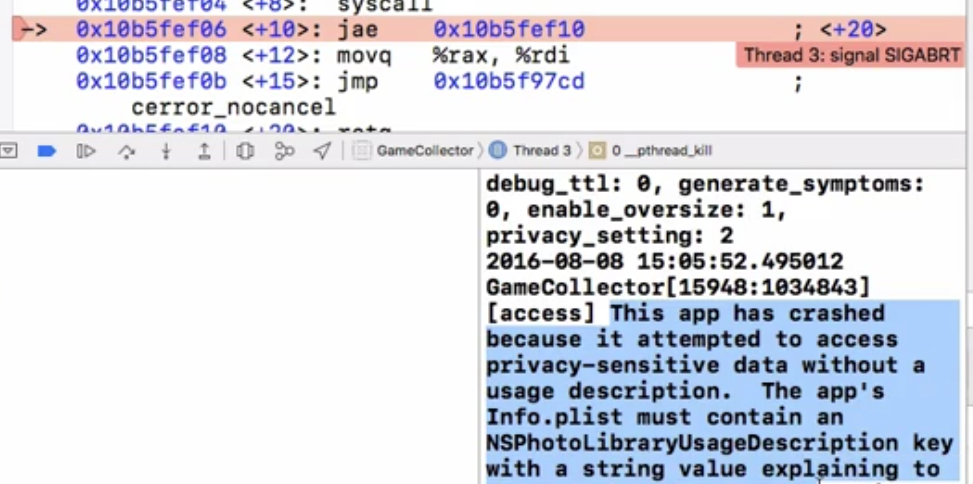
1. Cree un objeto de tipo ImagePickerController como variable de clase y delegue a la misma clase a ser controladora:



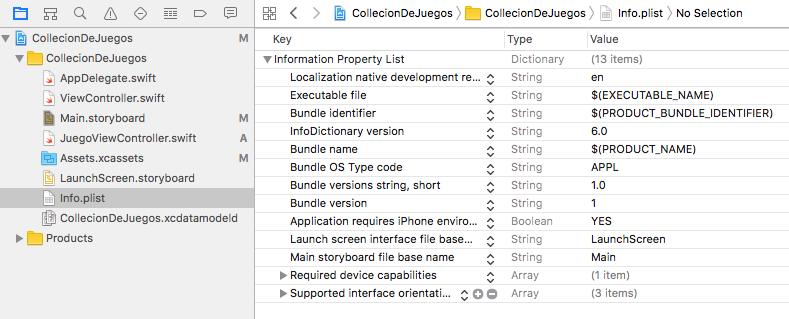
1. Implemente la función fotosTapped de la siguiente manera:



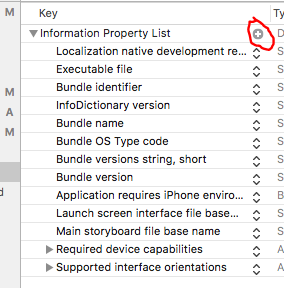
1. Ejecute la aplicación y obtendremos un error por los permisos que la aplicación no ha gestionado para usar las imágenes del dispositivo.



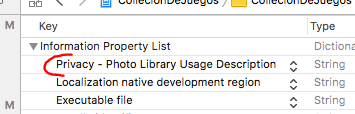
1. Visualice explorador de proyectos y visite el archivo **info.plist,** del proyecto.



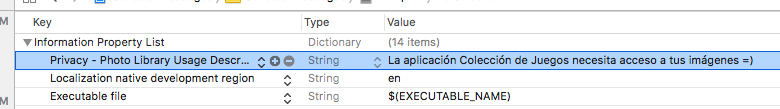
1. Clic en el botón de “+”



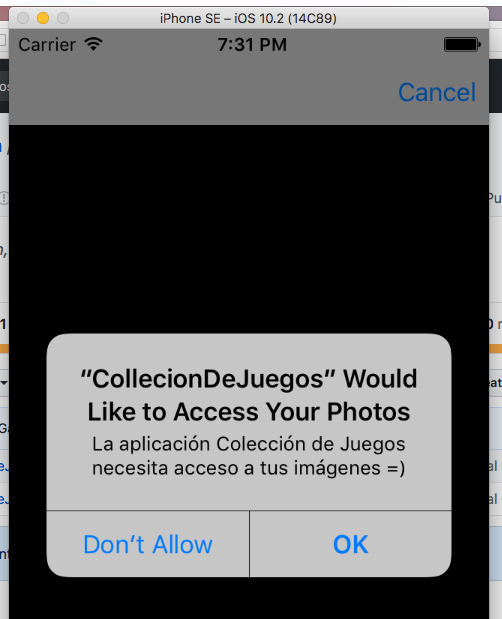
1. Busque y añada la categoría “***Privacy – Photo Library Usage Description***”



1. En la columna “value” complete: “**La aplicación Colección de Juegos necesita acceso a tus imágenes =)** ”

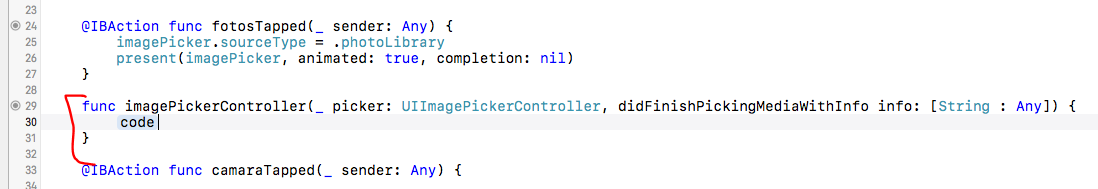


1. Ejecute la aplicación nuevamente y tap en el botón de “fotos” para abrir la galería del celular.

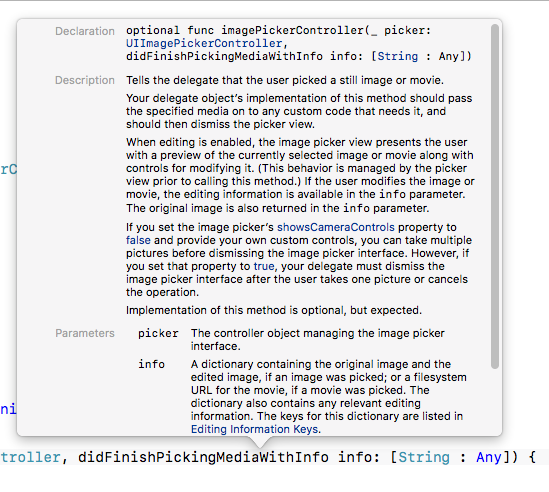


Aún no se seleccionará ninguna imagen, aún no se ha desarrollado esa porción de código pero debería poder explorarse la galería de imágenes

1. En el mismo controlador, implemente la siguiente función: (Función que se encargará de hacer alguna acción una vez seleccionada una imagen de la galería)



1. Para visualizar información de la función, mantenga presionada la tecla Alt + clic en el nombre del label argument “**didFinishPickingMEdiaWithInfo**”

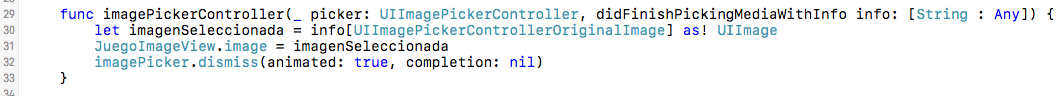


1. Para continuar obteniendo información acerca del **diccionario recibido**, clic en el link “***Editing Information Keys***”, se abrirá una página en Safari.



Vamos a hacer uso de la función marcada.

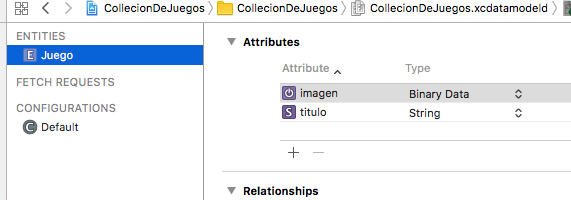
1. Modifique el código de la función para que quede de la siguiente manera:



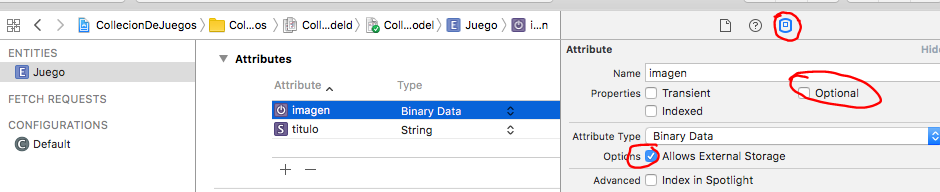
1. Ejecute la aplicación y compruebe que la imagen seleccionada se carga en el ImageView.

**CORE DATA**

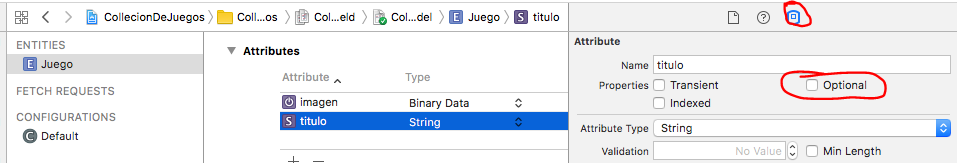
1. Agregue 1 entidad con las siguientes características:
   1. Nombre: Juego
   2. Atributos: 2
      1. titulo: String
      2. imagen: BinaryData



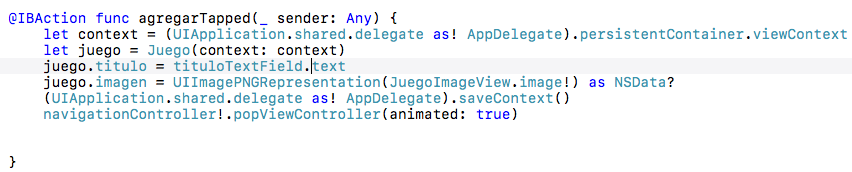
1. Seleccione el atributo ***imagen***, y al costado derecho, en las propiedades del atributo
   1. **Coloque Check** en Allows External Storage
   2. **Quite el Check** en Optional



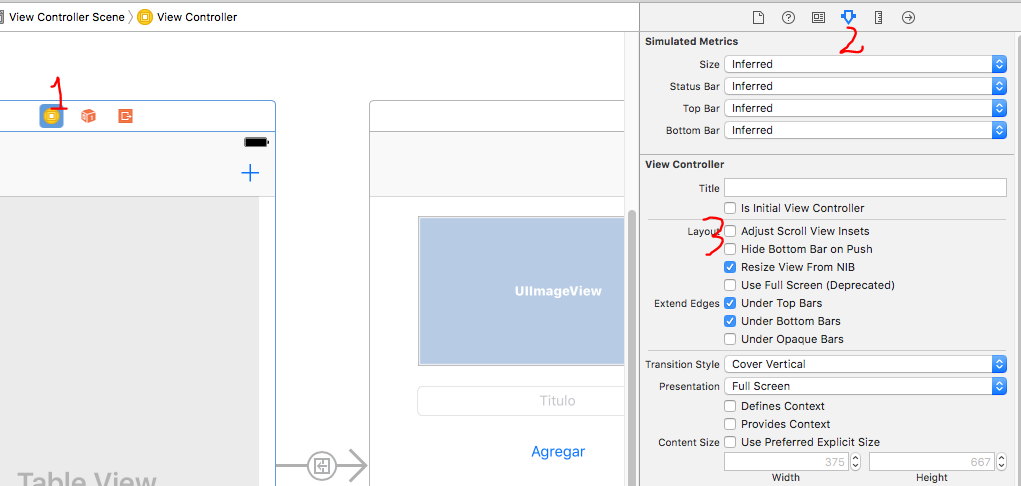
1. Seleccione el atributo ***titulo,*** y al costado derecho, en las propiedades del atributo:
   1. **Quite el Check** en Optional



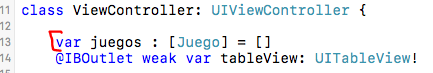
1. Regrese al controlador **JuegoViewController,** e implemente la función **agregarTapped**.



1. Diríjase al archivo main.storyboard y arrástre un control TableView al primer ViewController.
2. Añada constraints al TableView para que tenga 0 (cero) de margen con los elementos de todos sus lados.
3. Clic en el ícono del ViewController Principal y en sus propiedades quite el check de “**Adjust Scroll View Insets**”



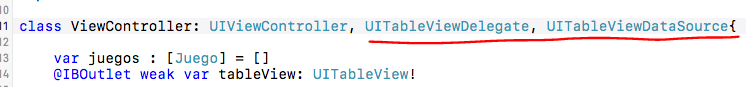
1. Ingrese al archivo ViewController (el principal).
2. Cree una conexión de tipo Outlet como variable de clase del TableView con nombre “**tableView**”.
3. Cree un arreglo de Entidades (Juego) como arreglo de clase de la siguiente forma.



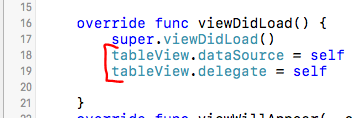
1. Borre la función **didReceiveMemoryWarning.**
2. Implemente la función **viewWillAppear** de la siguiente manera:



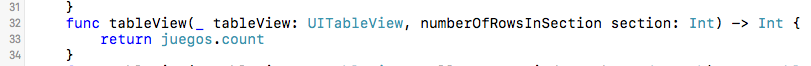
1. Agregue la funcionalidad necesaria para controlar a un TableView en el mismo controlador:
   1. Implemente los protocolos.



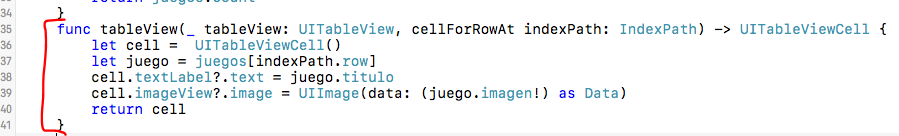
* 1. Asigne delegado y datasource



* 1. Implemente la función NumberOfRowsInSection



* 1. Implemente la función CellForRowAt

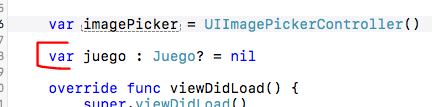


1. Compruebe que la aplicación funcione para agregar fotografías de la galería, descripción de las mismas y la visualización en el tableView del ViewContorller principal.
2. Haga commit en source tree para guardar los cambios.
   1. Clic en el botón de Commit
   2. Opcionalmente llene su nombre y correo asociado al commit
   3. Luego coloque la descripción de los cambios realizados.
   4. Finalmente, para cargarlo a GitHub, clic en el botón de **Push.**
   5. Cerciorarse de que los cambios se hayan efectuado en su cuenta de GitHub

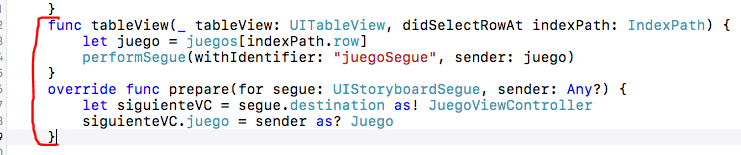


**CONFIGURACIÓN DE SEGUE**

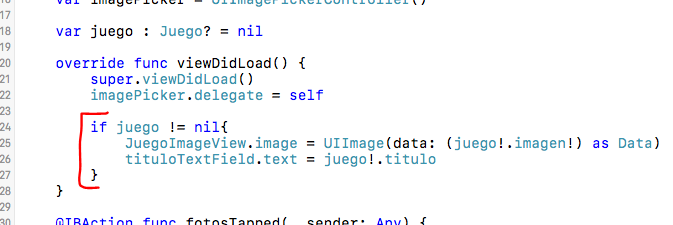
1. Seleccione el Segue que conecta a nuestro ViewController principal con el JuegoViewController.
2. Asigne el valor de su identificador como: “juegoSegue”
3. En el ViewController principal, implemente la función didSelectRowAt
4. Antes de inyectar código en la función creada en el paso anterior, regrese al JuegoViewController y cree una variable de clase de nombre **juego** con valor inicial **nil** de la clase **Juego.**



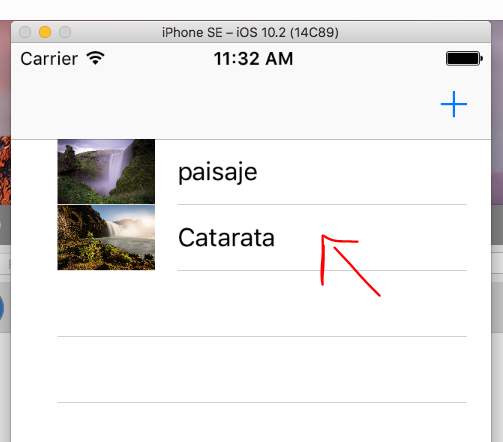
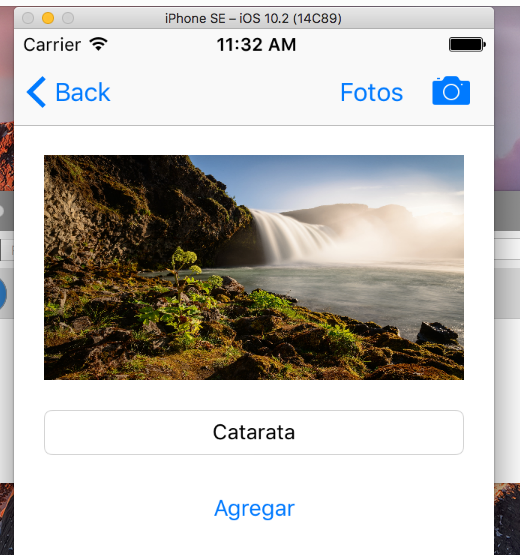
1. Ahora inyecte código en la función **didSelectRowAt** del ViewController principal e implemente la función **prepare** como indica la imagen a continuación:



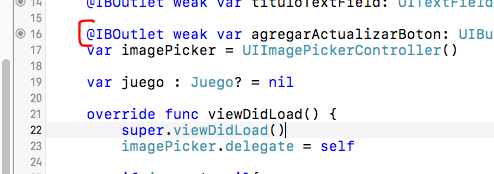
1. Regrese al **JuegoViewController** y modifique su función **viewDidLoad()** como indica la imagen a continuación.



1. Compruebe el funcionamiento de la aplicación.
   1. Se debe poder agregar fotos.
   2. Visualizar las fotos inmediatamente después de agregarlas en el TableView
   3. Y cuando se hace Tap en cualquiera de las imágenes, se debe regresar al JuegoViewController con la imagen seleccionada cargada y el título también.

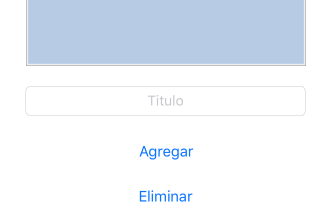
1. Anote los detalles más importantes:
2. Divida la pantalla para ver el archivo JuegoViewController y el main.Storyboard
3. Cree una conexión de **tipo OUTLET** del botón agregar a su ViewController como variable de clase con el nombre **agregarActualizarBoton.**



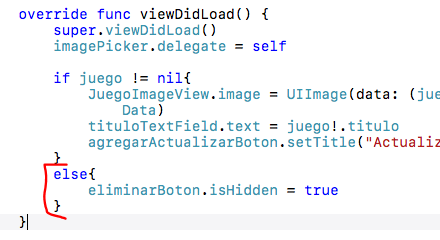
1. Agregue código a la función viewDidLoad del mismo controlador para que quede como en la siguiente imagen.



1. Ejecute nuevamente la aplicación y compruebe que cuando haga tap en una imagen el texto del botón cambie a “Actualizar”
2. Anote los detalles más importantes.
3. Abra el main.Storyboard y agregue un botón debajo del botón agregar con los siguientes constraints:
   1. Texto: “Eliminar”.
   2. Centrado Horizontalmente.
   3. Espacio vertical dependiente del botón Agregar con valor de 15



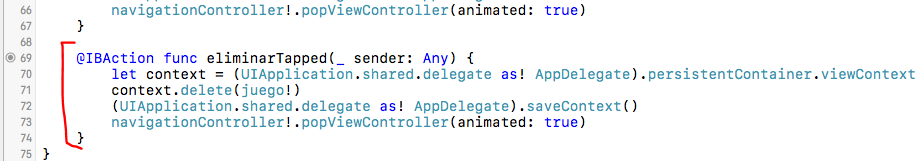
1. Cree un enlace de **tipo outlet** del botón eliminar a JuegoViewController como variable de clase con el nombre de: “eliminarBoton”.
2. Agregue código a la función viewDidLoad del JuegoViewController como se muestra a continuación:



1. Modifique la función agregarTapped para validar la acción de agregar o actualizar un elemento:

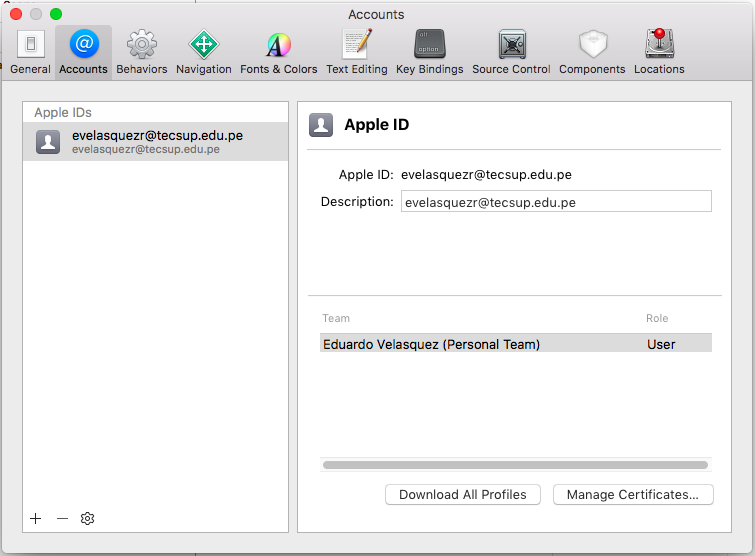


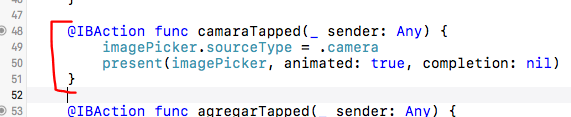
1. ¿Cuál es la diferencia entre la variable ***juego*** de la sentencia if con la variable ***juego*** de la sentencia else?¿Por qué tienen diferente color? Y cuál es la lógica que se utilizar para la validación de la acción agregar o actualizar
2. Compruebe ahora que la imagen se actualice y se agregue según sea la acción.
3. Cree una conexión de **tipo action** del botón **eliminar** hacia el **JuegoViewController** con el nombre de **eliminarTapped,** e inyecte código para que la función se vea de la siguiente manera:



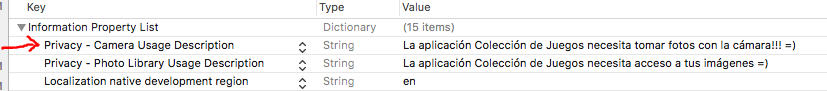
1. Ejecute la aplicación y compruebe que puede eliminar un juego.

**USANDO LA CÁMARA**

1. Hagaclic en el menú **Xcode -> Preferences.**
2. En la pestaña Accounts, asegúrese de estar *logeado* con su Apple ID en caso contrario agregue su cuenta con el botón +.
3. Cierre el cuadro de diálogo.
4. Abra el archivo JuegoViewController e inyecte código en la función **camaraTapped.**



1. Abra el archivo **info.plist**
2. Agregue un permiso para usar la cámara.



1. Para probar el correcto funcionamiento de la aplicación, conecte el iPhone a la MAC.
2. Elijar como dispositivo de renderizado el iPhone.
3. Ejecute la aplicación.
4. Anote los detalles más resaltantes.

**OBSERVACIONES (5 COMO MÍNIMO)**

**CONCLUSIONES (5 COMO MÍNIMO)**