

Desarrollo de aplicaciones para dispositivos móviles iOS

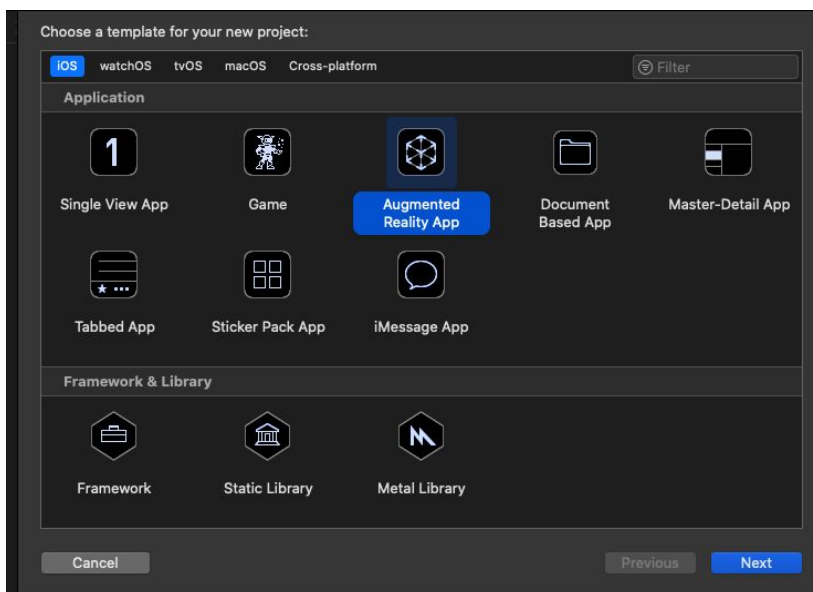
Tema 2 - Entorno de desarrollo integrado Xcode

INSTRUCTOR: Eduardo García Solís
lalogarciasolis@gmail.com

Marco contextual - Xcode

- Xcode es el ambiente de desarrollo integrado (IDE - integrated development environment) oficial del ecosistema Apple
- Desarrollado por Apple y lanzado en 2003. Versión actual 11.3.
- Para iOS, macOS, tvOS, iPadOs y watchOS

2.1 Tipos de aplicaciones¹



- Single View App: plantilla mínima y básica para empezar desde cero una aplicación iOS.

- Game: plantilla para desarrollar un juego.

- Augmented reality app: plantilla que vienen con la configuración necesaria para empezar a usar el kit de realidad aumentada de los dispositivos iOS.

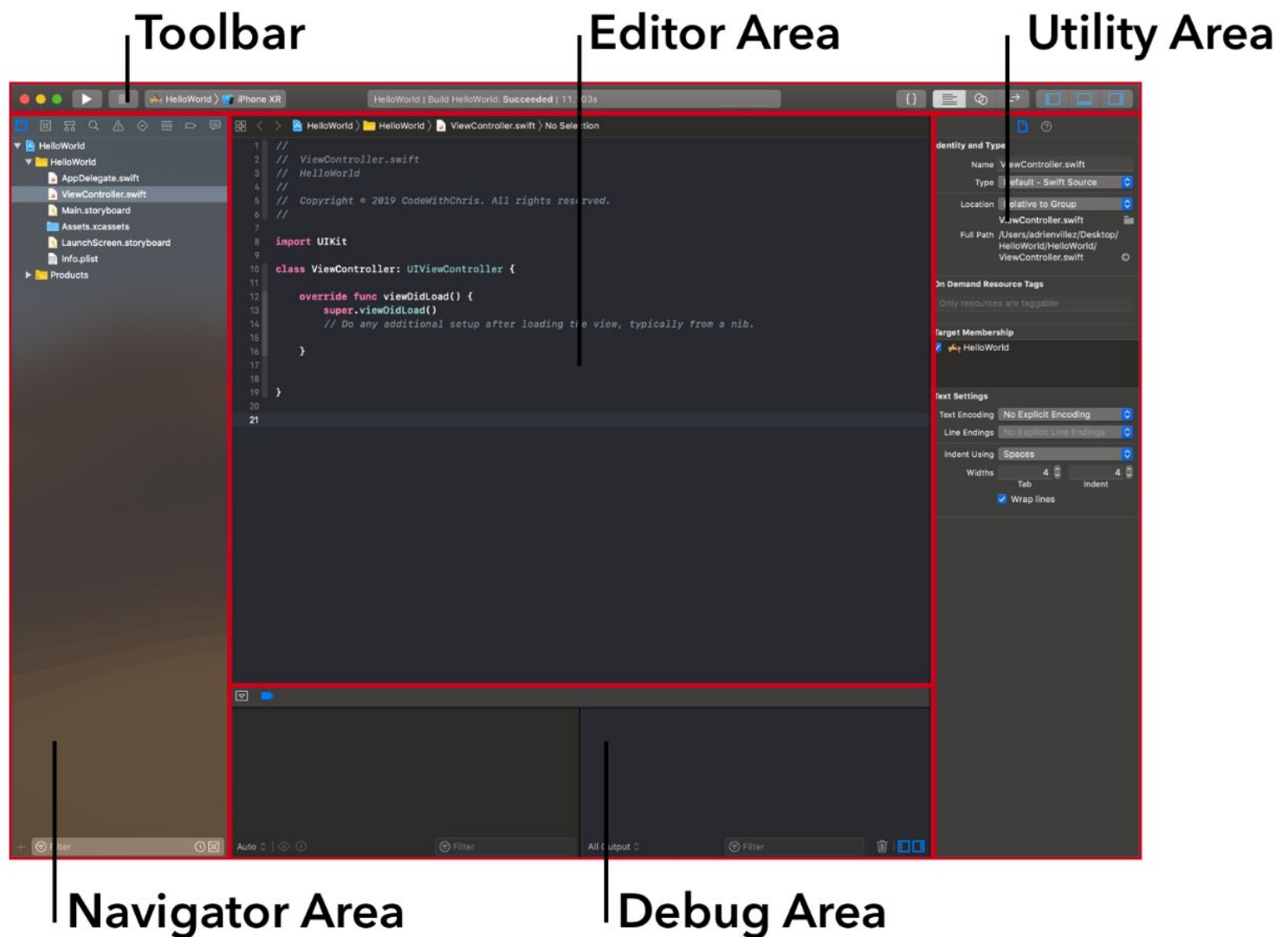
- Document Based App: plantilla para una aplicación que pueda crear, visualizar, editar, etc. diferentes tipos de

documentos.

- Master Detail-App: plantilla con elementos básicos de navegación de una aplicación clásica de iOS.
- Tabbed app: plantilla que viene de inicio con una Tab Controller (barra inferior de una aplicación)
- Sticker pack app: plantilla para empaquetar sticker para iMessage
- iMessage app: plantilla para una aplicación que se integra a iMessage

2.2 Áreas de trabajo²

Son las secciones principales que integran el ambiente de desarrollo que presenta Xcode.



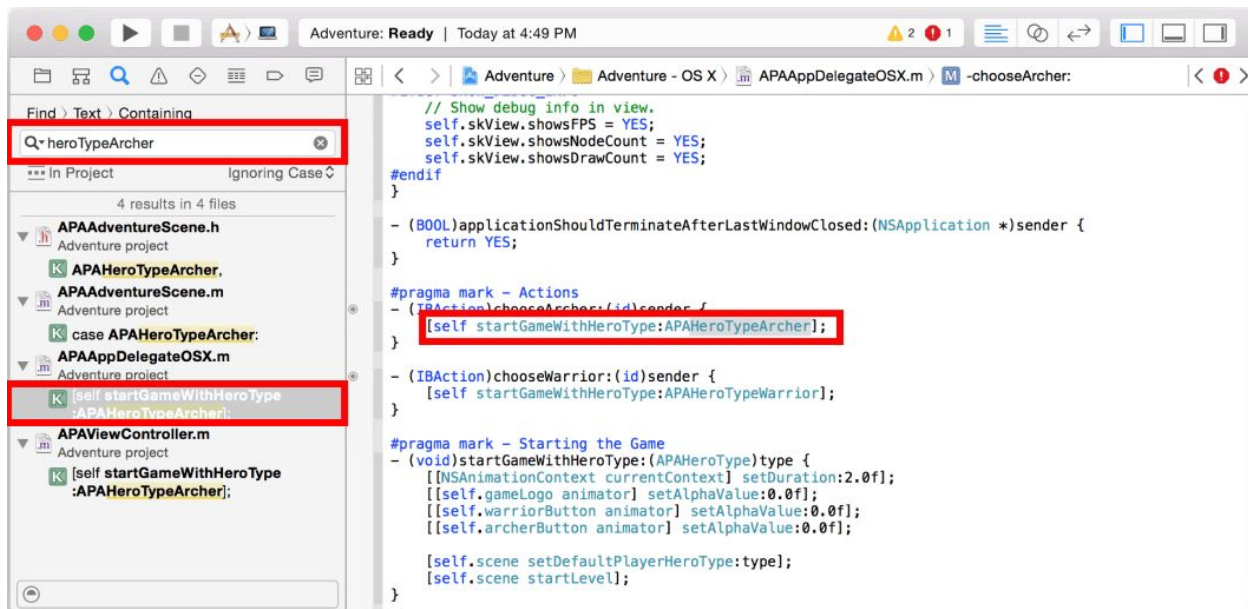
2.2.1 Navegación

Área desde la cual se puede acceder a archivos, símbolos, pruebas unitarias, diagnósticos y otras facetas de tu proyecto. Desde aquí se pueden organizar, crear o posibilitar la edición de archivos. De acuerdo a tu selección en el área de navegación dependerá lo que se mostrará en el área de edición.

- **Project navigator.** Add, delete, group, and otherwise manage files in your project, or choose a file to view or edit its contents in the editor area.
- **Symbol navigator.** Browse the symbols in your project as a list or hierarchy. Buttons on the left of the filter bar let you limit the shown symbols to a combination of only classes and protocols, only symbols in your project, or only containers.
- **Find navigator.** Use search options and filters to quickly find any string within your project.
- **Issue navigator.** View issues such as diagnostics, warnings, and errors found when opening, analyzing, and building your project.
- **Test navigator.** Create, manage, run, and review unit tests.
- **Debug navigator.** Examine the running threads and associated stack information at a specified point or time during program execution.
- **Breakpoint navigator.** Fine-tune breakpoints by specifying characteristics such as triggering conditions.
- **Report navigator.** View the history of your build, run, debug, continuous integration, and source control tasks.

2.2.2 Edición³

La mayoría del desarrollo ocurre en la área de edición, el área principal que siempre está visible dentro del espacio de trabajo.



Los editores que más se usan son:

- Editor de código
- Constructor de interfaces

- Editor de proyecto

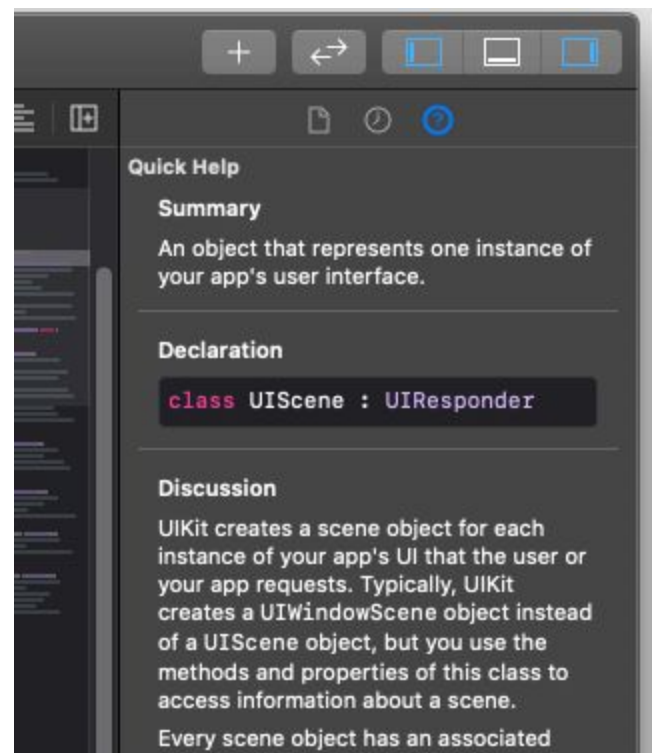
Como se dijo anteriormente, cuando tu seleccionas un archivo en el área de navegación, está se abrirá y se mostrará con el editor adecuado en el área de edición.

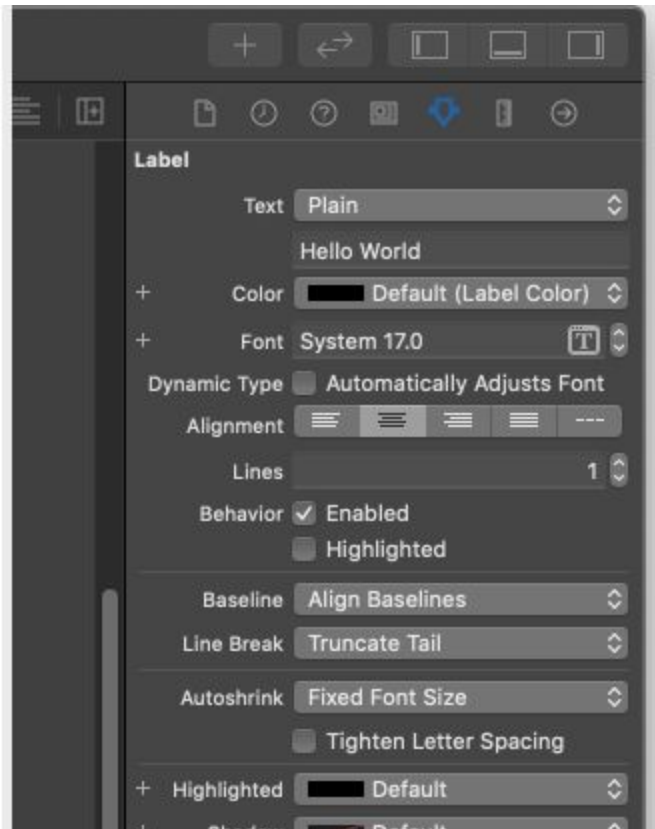
2.2.3 Utilería

La área de utilería se encuentra en el extremo derecho de Xcode, da un acceso rápido a los diversos inspectores los cuales sirven para visualizar y modificar características del archivos abierto. Los principales inspectores son:

- Inspector de archivos: muestra detalles y atributos destacados del archivo seleccionado en el área de navegación y visualizado en el área de edición.

- Inspector de ayuda rápida: muestra la documentación acerca del método, clase o palabra reservada donde el cursor de escritura se encuentra posicionado.



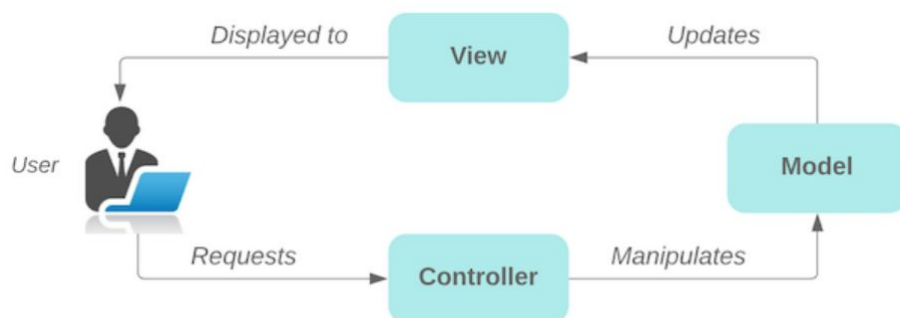


- Inspector de atributos e inspector de tamaño: estos inspectores aparecen cuando se está trabajando en un Storyboard (seleccionado en el área de navegación y mostrado en el área de edición), muestran y posibilitan la edición de los atributos visuales del componente UI seleccionado en el Storyboard.

2.3 Patrón Modelo - Vista - Controlador⁴

El patrón de diseño Modelo Vista Controlador (MVC) especifica que la aplicación consiste en 3 principales componentes:

- Modelo - Un modelo de datos: Representa la capa lógica de los datos y lógica de negocio. Los objetos de esta capa obtienen y almacenan el estado del modelo en la base de datos.
- Vista - Presentación de la información: Es la interfaz de usuario. Presenta los datos al usuario y también posibilita que este los modifique.
- Control - Administra de la información: se encarga de las peticiones del usuario, comúnmente el usuario interactúa con la vista la cual hace una solicitud que será manejada por el controlador. También se encarga de que la vista adecuada sea mostrada al usuario.

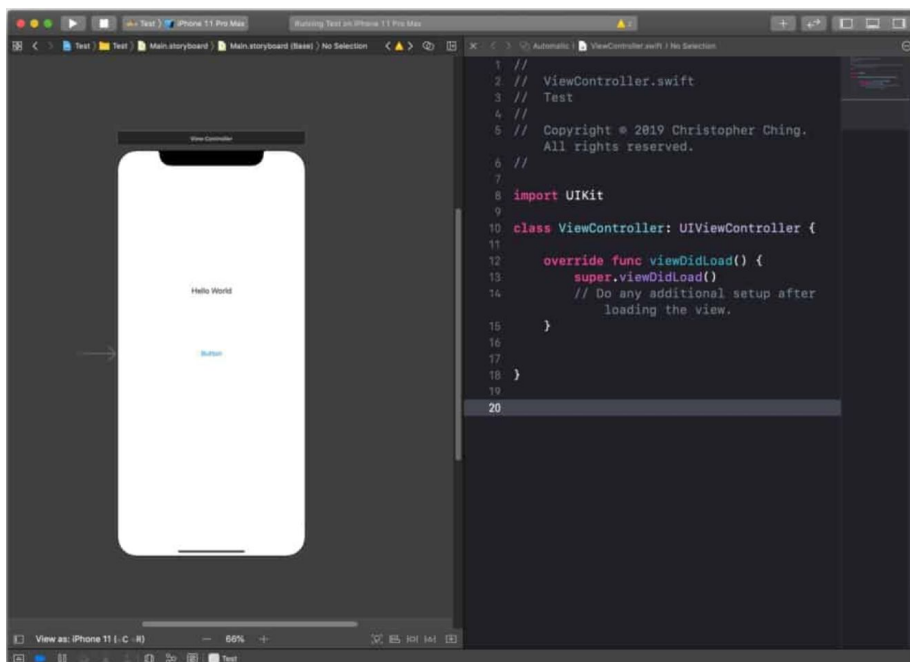
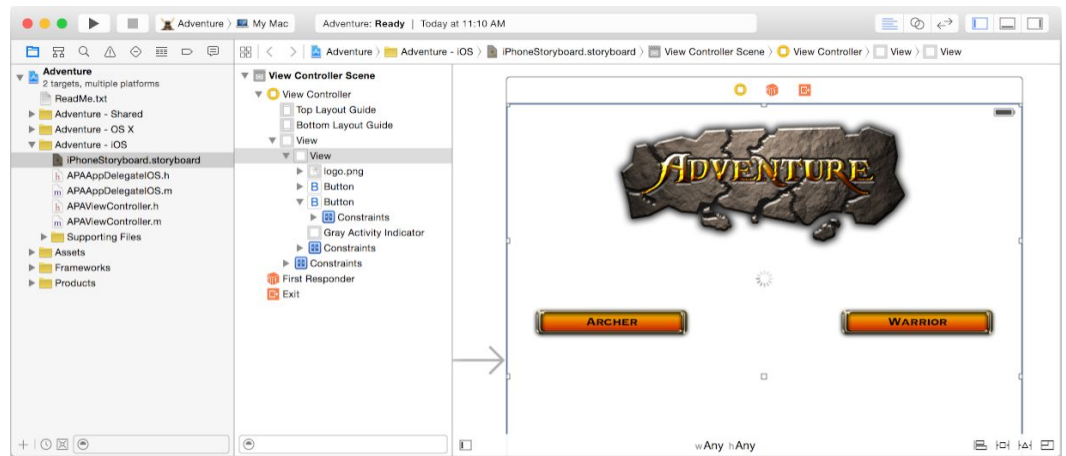


El patrón requiere la separación de cada una de estas capas y son representadas por diferentes objetos.

Este patrón de arquitectura ha existido por mucho tiempo en la ingeniería de software. Casi todos los lenguajes usan el MVC, con una ligera variación pero conceptualmente sigue siendo el mismo.

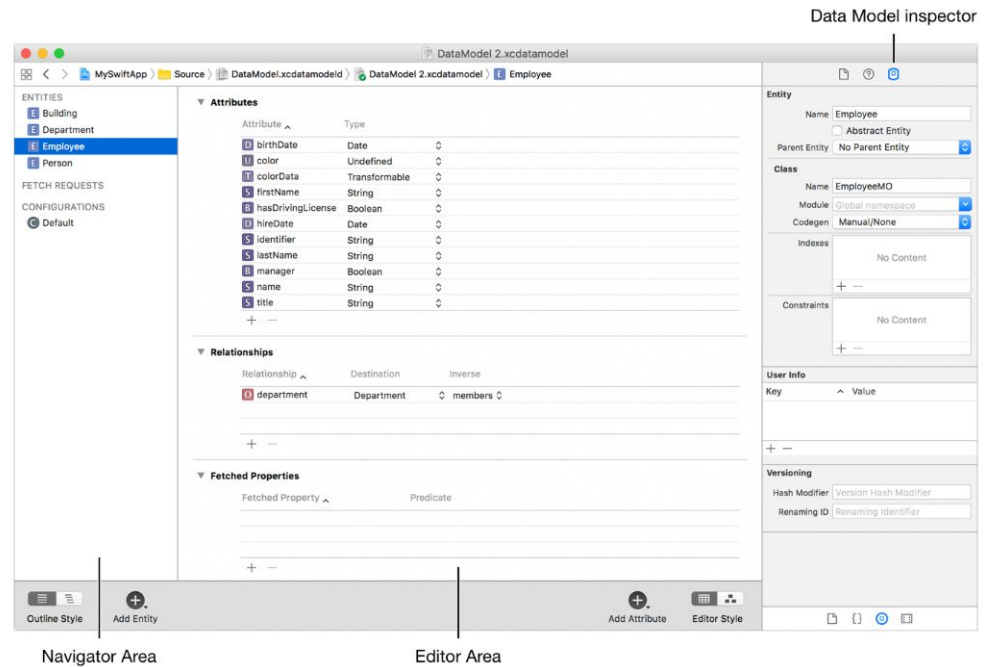
En una aplicación de iOS podemos identificar cada una de estas capas:

- Vista - Storyboard

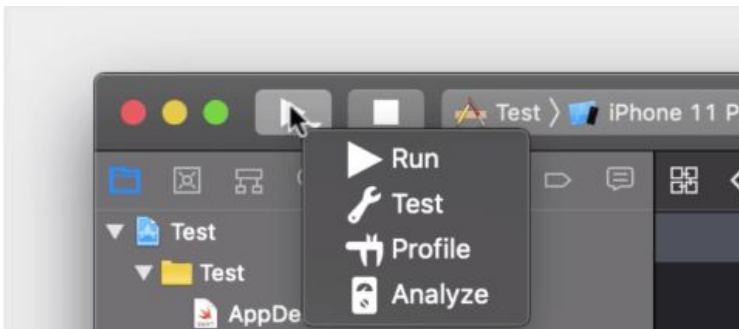


- Controlador - View Controller
(Clases controladores)

- Model - Clases entidad/CoreData



2.4 Implementación, ejecución y depuración⁶



En el extremo izquierdo y alto de Xcode podemos encontrar el botón Run, este botón construirá, ejecutará y lanzará la aplicación en un simulador de un dispositivo iOS o bien podrá ejecutarla en un dispositivo conectado al equipo Mac. Al lado se encontrará el botón Stop.

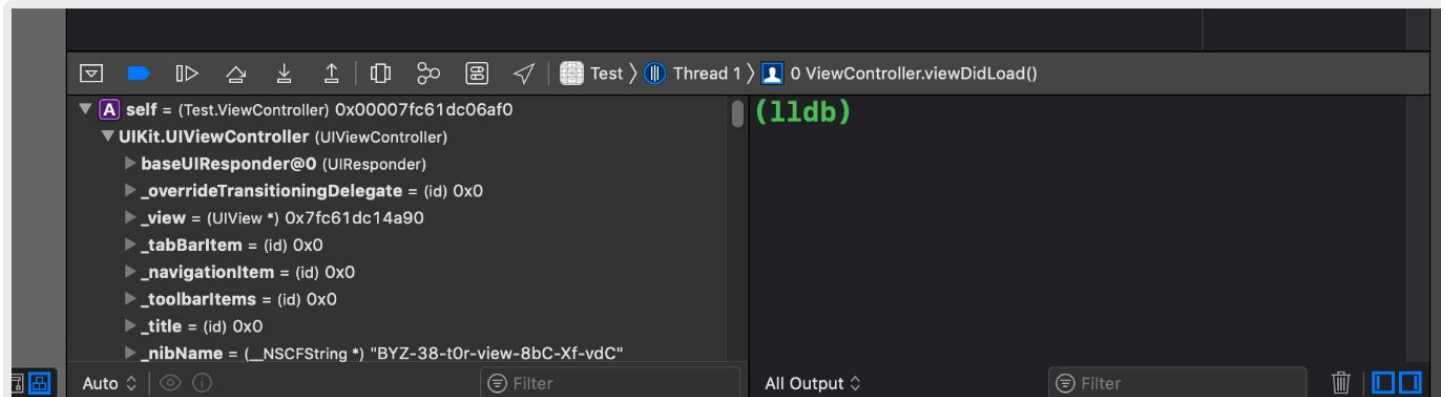
Podrás elegir el simulador de tu elección en la barra desplegable al lado del botón Stop.



En la barra de Status se mostrará el estado actual de la aplicación, si la ejecución no es exitosa podrás apreciar indicadores de errores o advertencias.



El área de debug mostrará la consola de salida y el estado de diversas variables cuando tu aplicación esté en ejecución. De igual forma podrás controlar la ejecución, agregar observadores (watchers) y poder gestionar tus puntos de quiebre (breakpoints).



Referencias

1. https://developer.apple.com/library/archive/documentation/ToolsLanguages/Conceptual/Xcode_Oview/CreatingProjects.html#//apple_ref/doc/uid/TP40010215-CH31-SW1
2. https://developer.apple.com/library/archive/documentation/ToolsLanguages/Conceptual/Xcode_Oview/TheWorkspaceWindow.html#//apple_ref/doc/uid/TP40010215-CH25-SW1
3. <https://codewithchris.com/xcode-tutorial/>
4. <https://www.geeksforgeeks.org/mvc-design-pattern/>
5. <https://openclassrooms.com/en/courses/5684146-create-web-applications-efficiently-with-the-spring-boot-mvc-framework/6124741-follow-spring-mvc-architecture-to-turn-static-html-into-a-thymeleaf-template>
6. <https://codewithchris.com/xcode-tutorial/>