

Informe Final
Modalidad Práctica Empresarial



ENTRENADOR EN PLATAFORMA iOS 1.0

PRESENTA:

SANTIAGO GÓMEZ GIRALDO

ASESORES:

**JAIME ANDRES LAINO GUERRA - ASESOR
EXTERNO DIANA MARGOT LOPEZ HERRERA -
ASESOR INTERNO**

**UNIVERSIDAD DE ANTIOQUIA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE
SISTEMAS ANTIOQUIA - MEDELLÍN
2019**

Entrenador en plataforma iOS 1.0

Resumen

Toda capacitación es muy importante durante la vida laboral y profesional, ya que de esta depende el funcionamiento que debe tener el trabajador o grupo asociado, la capacitación es el punto fuerte para que haya un desarrollo exitoso. El adiestramiento o capacitación es lograr la fuerza y los buenos resultados que una organización espera ya que estos logran que el empleado pueda tener una solución rápida y efectiva a cualquier contrariedad que se le presente durante su ocupación en el área asignada o externa; Sin embargo algunos procesos de capacitación, como en el caso de Globant, dependen de una única persona que es la que se encarga de dar instrucciones, coordinar y evaluar éste proceso, y esto puede conllevar a múltiples problemas cuando la persona encargada no está presente, o cuando el personal a capacitar es muy numeroso, por ésta razón se propuso el Entrenador en plataforma iOS, el cual tiene como objetivo, aligerar e incluso remover toda la carga del Instructor de la empresa.

Introducción

Globant es una empresa líder en el desarrollo de Software e innovación. Que cuenta con más de 6000 empleados localizados en oficinas de Argentina, Brasil, US, Colombia, España, México, Uruguay, India, Chile y el Reino Unido.

Cuya misión es convertirse en una empresa líder de desarrollo de software a nivel mundial [1].

En Globant, soñamos y construimos viajes digitales que importan a millones de usuarios. Este tipo de viajes digitales excede la creación de un sitio web, una aplicación o incluso una experiencia omnicanal unificada. Implica la creación de una relación más profunda con los usuarios mediante la entrega de experiencias memorables que son personalizadas, sensibles al tiempo y al contexto y la ubicación. Es lo que llamamos una experiencia omni relevante.

Globant está comprometido con la satisfacción del cliente al continuar con la idoneidad y eficacia para lograr los objetivos de nuestra Política de calidad. Esta política proporciona la base para establecer los objetivos para evaluar la eficacia de nuestro Sistema de gestión de calidad [1].

Globant ha construido una lista de clientes de la clase mundial, muchos de los cuales están a la vanguardia de las tecnologías emergentes, entre los cuales se destaca Disney, una compañía multinacional de entretenimiento, para cual se desarrollan múltiples aplicaciones.

Con el fin de asegurar la calidad de los proyectos realizados para Disney, el Site de Globant en Medellín, realiza un proceso de entrenamiento básico en desarrollo de aplicaciones iOS para los practicantes que ingresan a la empresa cada semestre, este proceso es coordinado por el referente en iOS, el cual asigna tareas y las revisa conforme se completan.

Sin embargo, este tipo de proceso no está automatizado y se podría ver gravemente afectado cuando la persona encargada del entrenamiento no esté presente, igualmente todos los practicantes deberían tener un mínimo viable de conocimiento y competencias de trabajo en dicha plataforma, con el fin de garantizar la una capacidad unificada básica para iniciar el desarrollo de los proyectos.

Por esta razón se propuso el desarrollo del “**Entrenador en plataforma iOS 1.0**” el cual tiene como objetivo presentar una forma más personal y autodidacta de realizar este proceso de entrenamiento, sin necesidad de depender de un ente externo y que esta pueda evaluar de forma automática el progreso de cada estudiante durante esta etapa de entrenamiento. Igualmente este entrenador podrá actualizarse y robustecerse de acuerdo a los cambios de la plataforma iOS.

Objetivos

General:

Desarrollar una herramienta de entrenamiento básico, sobre el proceso de creación de aplicaciones para dispositivos iOS, haciendo uso de la herramienta Xcode y el lenguaje de programación Swift .

Específicos:

- Revisar los conceptos mínimos necesarios para la creación de una aplicación para dispositivos iOS.
- Seleccionar los conceptos y ejemplo necesarios para generar capacidad de desarrollo de aplicaciones en dispositivos iOS. en practicantes de Ingeniería de sistemas.
- Diseñar una guía instruccional, de los criterios de capacitación básicos, para la creación de una aplicación para dispositivos iOS usando Xcode y Swift
- Construir la capa FrontEnd que presente los diferentes contenidos y actividades de interacción del entrenador.
- Construir un componente evaluador del conocimiento y de la habilidad de desarrollo que ha logrado el practicante.
- Preparar los contenidos, ejemplos, código de ejemplo y actividades evaluativas que nutrirán el entrenador.

Marco Teórico

iOS es un sistema operativo móvil de la multinacional Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), después se ha usado en dispositivos como el iPod touch y el iPad. No permite la instalación de iOS en hardware de terceros.

Actualmente es el segundo sistema operativo móvil más utilizado del mundo, detrás de Android, con una cuota de mercado de entre 10-15% al año 2017. La última versión del sistema operativo es iOS 12 aparecida en junio de 2018 sustituye a iOS 11 con el objetivo principal de mejorar la experiencia del usuario [2].

Genéricamente debemos definir al Entrenamiento como toda práctica o conjunto de prácticas que nos lleven a adquirir un perfeccionamiento en un área determinada, tanto en nuestras Habilidades Físicas como también en lo que respecta a nuestras Habilidades Intelectuales (es decir, un entrenamiento ligado al Conocimiento), incrementando nuestras capacidades psicofísicas y poniéndonos a prueba de distintas habilidades, mejorando nuestras aptitudes para poder afrontar una adversidad específica o para realizar una tarea determinada.

Se define entonces al Entrenamiento como una especie de paralelismo a lo que es el Aprendizaje, teniendo que tener una base de conocimientos específica para poder realizarse, o bien pensar en un Objetivo que debe ser la base del punto de partida que se debe realizar, considerándose como tal no solo al relativo al Entrenamiento Físico, sino que también debemos tener en cuenta a la Formación Académica de un individuo [3].

La empresa Globant está comprometida con la satisfacción del cliente, por esa razón, los nuevos integrantes de la empresa pasan por diferentes procesos de entrenamiento respectivos al área que son asignados, que en nuestro caso es un entrenamiento enfocado en conocer los conceptos básicos de desarrollo de aplicaciones para dispositivos iOS, sin embargo, este proceso de entrenamiento es coordinado por un referente, el cual brinda indicaciones y revisa lo que se va haciendo, para así continuar con el siguiente paso, esto, a pesar de ser un proceso más completo personalizado, puede traer inconvenientes, ya que depende totalmente del referente, y en caso de que él no esté presente, el proceso se detendría totalmente; Por esta razón se propuso crear una plataforma de entrenamiento, haciendo uso de los mismos conceptos que se aprendieron en este, para así facilitar el proceso de aprendizaje y que no dependa de una persona externa al proceso [1].

En este proceso de entrenamiento se tienen varios objetivos:

- Aprender lo básico sobre desarrollo de aplicaciones iOS.
- Aprender a leer documentación y fomentar el autoaprendizaje.
- Uso básico de Git.
- Aprender sobre patrones de Diseño.
- Interacción con Xcode y Swift.
- Uso del Interface Builder de Xcode y el UIKit proporcionado por Apple.

El primer paso a la hora de desarrollar aplicaciones para iOS, es tener conocimiento sobre el “**Human Interface Guidelines**” una guía sobre las diferentes herramientas que visuales que nos proporciona Apple y cómo se deben utilizar, para que cumplan con los criterios de diseño de la compañía [4].

Después de tener conocimiento sobre el “**Human Interface Guidelines**” ya se puede hacer el primer contacto con Xcode, el cual es un entorno de desarrollo integrado (IDE) para macOS que contiene un conjunto de herramientas de desarrollo de software desarrolladas por Apple para desarrollar software para macOS, iOS, watchOS y tvOS. Sin embargo en este caso sólo nos enfocaremos en la parte de iOS.

Xcode soporta código fuente en los lenguajes de programación C, C ++, Objective-C, Objective-C ++, Java, AppleScript, Python, Ruby, ResEdit (Rez) y Swift, de los cuales, los más usados son Objective-C y Swift. Sin embargo Apple actualmente recomienda Swift, ya que es el más nuevo y fue diseñado directamente por ellos [5].

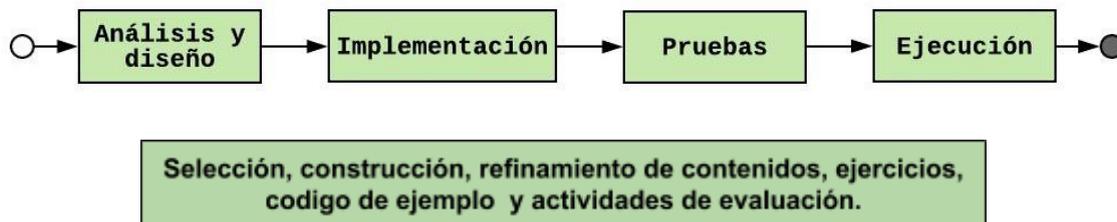
Mencionados los conceptos básicos a tener en cuenta, entraremos en detalle sobre las herramientas necesaria para desarrollar la aplicación “**Entrenador en plataforma IOS**”.

- **iOS:** iOS es un sistema operativo móvil creado y desarrollado por Apple Inc. exclusivamente para su hardware. Es el sistema operativo que actualmente impulsa muchos de los dispositivos móviles de la compañía, incluidos el iPhone, iPad y iPod Touch [2].
- **Xcode:** Xcode trabaja conjuntamente con Interface Builder, una herencia de NeXT, una herramienta gráfica para la creación de interfaces de usuario. Xcode incluye la colección de compiladores del proyecto GNU (GCC), y puede compilar código C, C++, Swift, Objective-C, Objective-C++, Java y AppleScript mediante una amplia gama de modelos de programación, incluyendo, pero no limitado a Cocoa, Carbón y Java[4].
- **Swift:** Swift es un lenguaje de programación multiparadigma creado por Apple enfocado en el desarrollo de aplicaciones para iOS y MacOS. Fue presentado en WWDC 2014 y está diseñado para integrarse con los Frameworks Cocoa y Cocoa Touch, puede usar cualquier biblioteca programada en Objective-C y llamar a funciones de C [6].
- **Alamofire:** Alamofire es una biblioteca de redes HTTP basada en Swift para iOS y macOS. Proporciona una interfaz elegante en la parte superior de la pila de red de Apple Foundation que simplifica una serie de tareas de red comunes [7].
- **CocoaPods:** CocoaPods es un administrador de dependencia para los proyectos en Swift y Objective-C Cocoa. Tiene más de 51 mil bibliotecas y se usa en más de 3 millones de aplicaciones. Este proporciona un formato estándar para administrar bibliotecas externas. Fue desarrollado por Eloy Durán y Fabio Pelosin, quienes continúan administrando el proyecto con la ayuda y las contribuciones de muchos otros [8].

Metodología

Para el proceso instruccional que tendrá este aplicativo se realizará la selección, construcción, refinamiento tanto de contenidos como de ejemplos y actividades, buscando generar dinámicas de aprendizaje amenas y pertinente con el objetivo de entrenamiento.

Para el análisis y desarrollo de la herramienta se definieron cuatro fases secuenciales y una fase transversal que abarcan los pasos que se consideran necesarios para alcanzar el objetivo, las fases son las siguientes:



- **Análisis:** Durante esta fase se define la arquitectura y las tecnologías a usar. Además, se realizan las respectivas mockups.
- **Implementación:** Consta de un release con 5 sprints definidos de la siguiente manera:
 - **Sprint 1:**
 - Definición de la información que contendrá el curso.
 - Implementación de un Servicio Web que contenga dicha información.
 - Implementación de una TableView
 - Asignar el contenido del servicio web en la TableView
 - **Sprint 2:**
 - Implementación de un Servicio Web que contenga información específica de cada tema.
 - Implementar una DetailView
 - Desplegar los detalles de cada tema en el DetailView cuando el usuario haga tap en la celda.
 - **Sprint 3:**
 - Dentro del DetailView crear un Botón que despliega una nueva vista con un ejercicio propuesto sobre el tema actual.
 - Crear otra vista que despliega una posible solución al ejercicio propuesto.
 - Crear otra vista que despliegue cómo se ve en la aplicación la solución propuesta.
 - Crear un TabBar que permite cambiar entre las tres vistas

- **Sprint 4:**
 - Crear una vista que despliegue una pregunta sobre el tema que se está visualizando.
 - Implementar la funcionalidad del cuestionario.
- **Sprint 5:**
 - Crear una barra de progreso que indique que porcentaje del entrenamiento está completo.
 - Implementar pruebas.
 - Refactorizar el código.
- **Pruebas:** Se realizan pruebas de aceptación con el propósito de mejorar y estabilizar la herramienta.
- **Ejecución:** Esta fase es transversal a las demás, puesto que acá se elabora la documentación.

En cuanto al desarrollo del aplicativo en La fase de implementación se utilizará la metodología de desarrollo ágil Scrum, la cual es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente en equipo y obtener el mejor resultado posible de un proyecto [9]. En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite máximo de feedback y reflexión). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite [9].

Es importante mencionar que no se estará completamente apegado a ésta metodología puesto que por cuestiones de tiempo y porque no se tiene un equipo con los diferentes roles que se requieren, no se realizarán todas las ceremonias que se plantean en Scrum.

Resultados y análisis.

El objetivo principal de esta aplicación era facilitar el proceso de capacitación de los practicantes que ingresaban a la empresa, por esto se utilizaron los conceptos mínimos necesarios para la creación de una aplicación para dispositivos iOS presentados por el entrenador de la empresa, los cuales son:

- **Human Interface Guidelines:** Se proporcionaron las guías disponibles en la documentación de Apple.
- **Xcode:** Se hace una breve explicación del entorno y se incita a la exploración de este.
- **Swift:** Se hace una breve explicación del lenguaje y se proporcionan ejemplos de las estructuras básicas.
- **UIViewController:** Se proporciona la documentación de Apple, además de ejemplos sencillos.
- **Alamofire:** Se explica brevemente el uso de la librería y se proporcionan ejemplos básicos.
- **Cocoapods:** Se explica el objetivo del manejador de dependencias y se muestra un ejemplo.
- **UITableViewController:** Se proporciona la documentación de Apple, además de ejemplos sencillos.
- **UICollectionViewController:** Se proporciona la documentación de Apple, además de ejemplos sencillos.

Debido a que la aplicación que se presentará a continuación es un POC (Proof of concept), toda la información mencionada anteriormente no se encuentra en la APP. Se considera entonces que la información conceptual se consigne en un documento anexo que el practicante pueda consultar asincrónica o sincrónicamente con la App de acuerdo con su estilo de aprendizaje. Igualmente este documento se convierte en un activo de conocimiento para el entrenamiento de practicantes que puede evolucionar de acuerdo a los cambios que se generan desde la casa matriz y el mismo iOS. Por cuestiones de seguridad el detalle de la información no puede presentarse, se anexa la tabla de contenido para mostrar la extensión y temáticas abordadas. Igualmente el asesor Externo da fé que la tabla de contenido que aquí se relaciona se ha desarrollado y que el documento está completamente terminado.

Table Of Contents

iOS Training	4
Overview	4
Goals	4
Specifications	4
Expectation	4
Milestones	4
Basic Understanding	4
Fully functional Utility App	4
iOS Human Interface Guidelines	5
Overview	5
Goals	5
Specifications	5
The App Life Cycle	6
Overview	6
Goals	6
Specifications	6
UIViewController	7
Overview	7
Goals	7
Specifications	7
Software design pattern	8
Overview	8
Goals	8
Specifications	8
Model-View-Controller	9
Overview	9
Goals	9
Specifications	9
The Movie DB (App)	10
Overview	10
Goals	10



Specifications	10
GIT & Github	11
Overview	11
Goals	11
Specifications	11
Networking	12
Overview	12
Goals	12
Specifications	12
JSON & Parsing	13
Overview	13
Goals	13
Specifications	13
Facade Pattern	14
Overview	14
Goals	14
Specifications	14
Memory Management	15
Overview	15
Goals	15
Specifications	15
UITableView	16
Overview	16
Goals	16
Specifications	16
UICollectionView	17
Overview	17
Goals	17
Specifications	17
Detail View	18
Overview	18
Goals	18
Specifications	18
SOLID (object-oriented design)	19
Overview	19



Goals	19
Specifications	19
Single responsibility principle	20
Overview	20
Goals	20
Specifications	20
Test-driven development	21
Overview	21
Goals	21
Specifications	21
Unit Testing	22
Overview	22
Goals	22
Specifications	22

Entrenador iOS - Guía de uso

Visión General

Este entrenamiento tiene como objetivo, enseñar los conceptos básicos para el desarrollo de aplicaciones iOS, cosas como, El ciclo de vida de una aplicación, Lineamientos de interfaz humana, Networking, Patrones de Diseño, Sistemas de control de versionamiento y más; Todo esto haciendo uso de Xcode y Swift.

Metas

1. Adquirir conocimiento básico sobre desarrollo iOS.
2. Incentivar la lectura de documentación y autoaprendizaje.
3. Uso Básico de Git.
4. Aprender sobre Patrones de Diseño.
5. Interactuar con Xcode y Swift.
6. Uso del "Interface builder" y UIKit.

Uso

Al estudiante se le proporcionará acceso al documento y al aplicativo de forma manual, el objetivo es que trabaje cada tema en el orden que están planteados, cada tema contiene información teórica, además de links que permiten profundizar la información, después de tener los conceptos teóricos claros se propone un ejercicio práctico que el practicante puede desarrollar en su equipo asignado, para apoyar esto, se proporciona un ejemplo de la implementación del ejercicio propuesto en la app; Finalmente dentro de la app debe responder correctamente una pregunta técnica y acto seguido continuar con el siguiente tema.

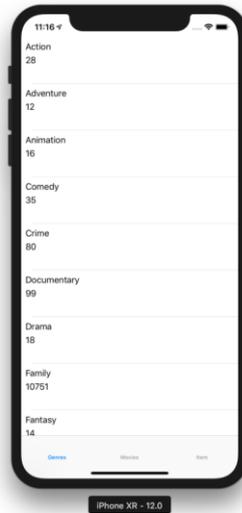
Para tener un proceso de entrenamiento fluido se recomienda tratar de 2 a 3 temas semanalmente.

Resultado esperado

Se espera que el entrenamiento tome máximo 2 meses y que al final de este el Practicante tenga un conocimiento de nivel "Junior" o "Junior Advance" y que tenga todas las herramientas para seguir investigando y mejorando de forma autónoma.

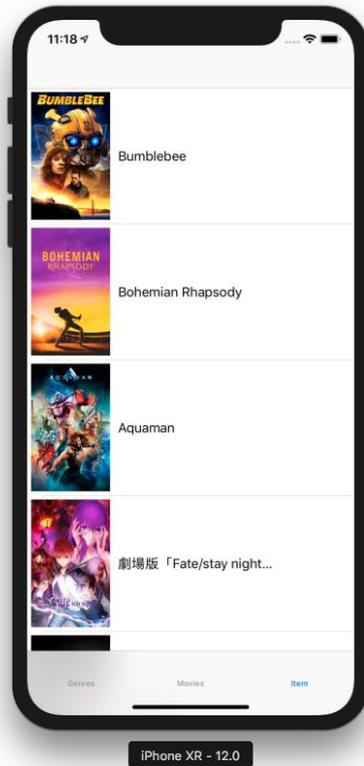
Con el fin de dotar el activo de conocimiento del Saber (documento) con una parte del Hacer, para los diferentes conceptos se plantearon ejemplos que tenían como objetivo generar una capacidad de desarrollo (hacer) en los practicantes, algunos de estos son:

- **UIViewController:** Se plantea crear una vista simple haciendo uso de la clase proporcionada por Apple, ejemplo:

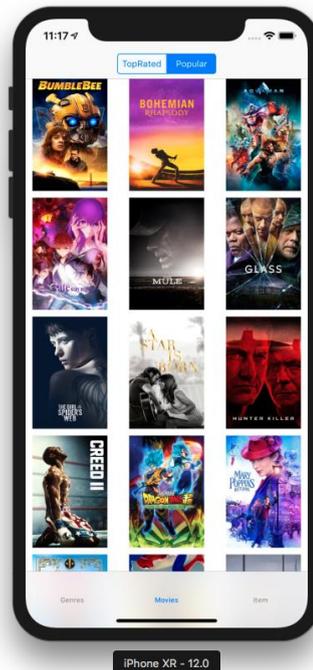


- **Software Design Patterns:** En este caso se presentan diferentes patrones de diseño básicos y se propone un ejemplo haciendo uso del patrón de Diseño Singleton.
- **Model-View-Controller:** Se presenta la arquitectura Modelo-Vista-Controlador y se propone distribuir el código de la vista, los datos y la lógica de negocio en diferentes clases.
- **Github:** Se hace una introducción a la plataforma, se explica cómo la utilizan en la empresa y se propone crear un repositorio, hacer push en el repositorio remoto, después crear una branch, hacer un cambio y finalmente hacer un merge entre la branch con el cambio y la versión de master.
- **Networking:** Se explica brevemente en qué consisten los servicios web y se plantea un ejercicio de prueba haciendo un llamado a un servicio aprobado por la empresa.
- **JSON & Parsing:** Se explica en qué consiste el formato JSON y se plantea un ejercicio en el que se convierte un objeto tipo JSON, en diferentes objetos primitivos de Swift.
- **Memory Management:** Se presenta la forma en la que Swift hace gestión de la memoria y se proponen ejemplos utilizando, Strong, Weak y Unowned, conceptos que están explicados en la guía.

- **UITableView:** Se plantea crear una tabla simple haciendo uso de la clase proporcionada por Apple, ejemplo:



- **UICollectionView:** Se plantea crear una vista simple haciendo uso de la clase proporcionada por Apple, ejemplo:



- **Test-driven development:** Se presenta el concepto y se propone un ejemplo en el que primero se debe crear la prueba, después debe hacer que la prueba sea exitosa de manera forzada para un caso específico y finalmente se hace un refactor del código para que apruebe en cualquier caso.

Cuando un desarrollador domine todos estos conceptos se espera que ya tenga un nivel de al menos Junior o Junior Advance dentro de la compañía. Se espera que con un estudio de al menos 4 horas diarias este nivel Junior se alcance en 60 días.

La parte de la App como activo de conocimiento, el resultado fue una herramienta con las funcionalidades pactadas desde un inicio. La idea es que el practicante tenga el documento donde podrá estudiar y ver los ejemplos y en la aplicación.

Se usaron todas las tecnologías definidas a excepción de los servicios web propios, que debido a políticas de la empresa no se podían montar y llamar servicios web diferentes a los autorizados, por esa razón se tuvo que almacenar información de manera local. Se puede decir que este fue el mayor inconveniente presentado durante las diferentes fases del proyecto.

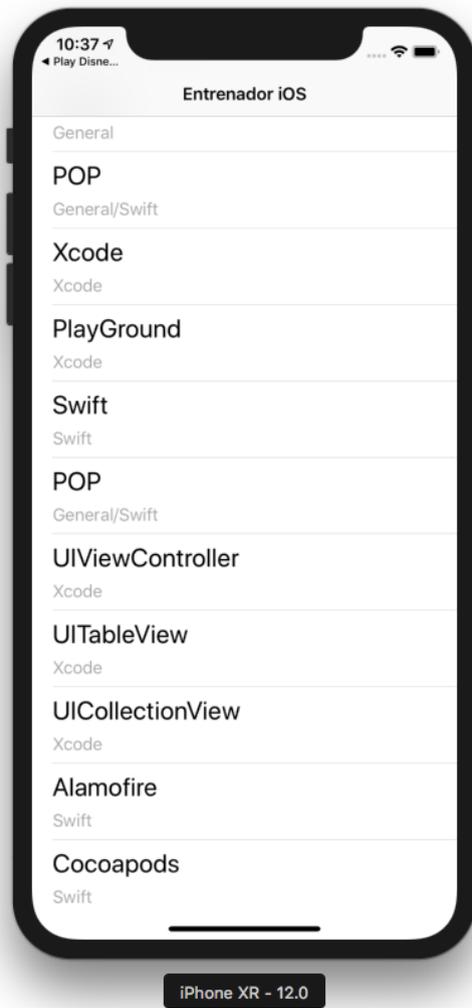


Figura 1. Pantalla inicial de la aplicación

En la **Figura 1** se puede observar la pantalla inicial de la aplicación, la cual contiene una lista con los diferentes temas que son importantes a la hora de desarrollar una aplicación en iOS y a los cuales se puede acceder en esta aplicación.

Para esta vista se utilizó una “TableView”, que es una clase proporcionada por el UIKit de Apple y la cual también está incluida en la aplicación, el objetivo de esta clase es crear una tabla con una columna y múltiples filas en la que se implementa el patrón de Diseño “Prototype”, creando el prototipo de las celdas y acto seguido se multiplican y se llenan con la información necesaria tal como se muestra en la **Figura 2**.

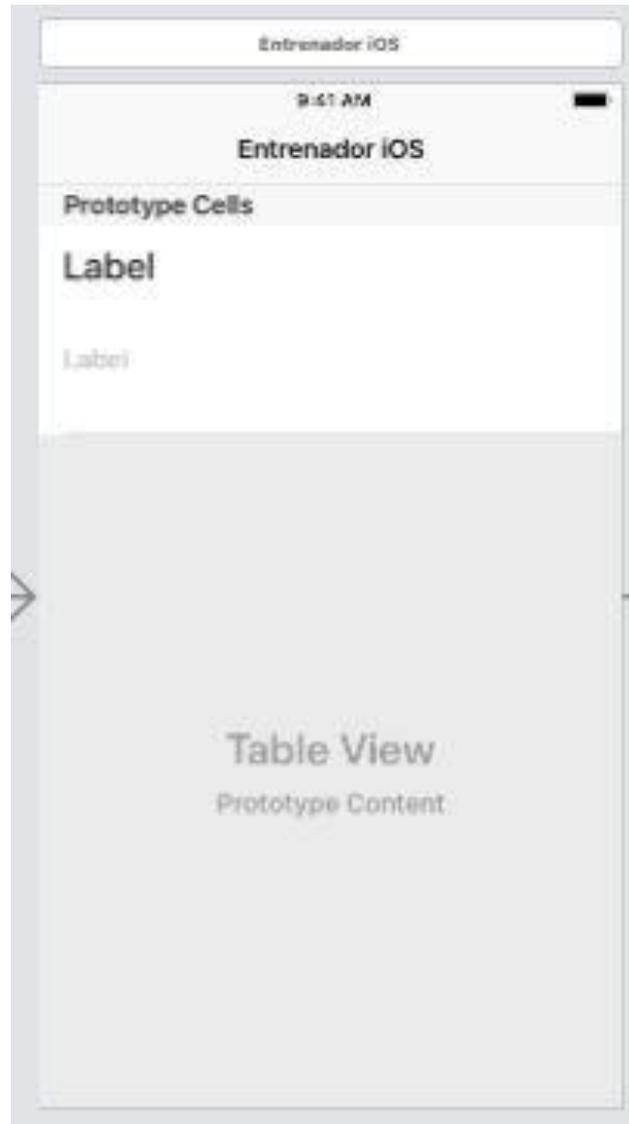


Figura 2. Prototipo de las celdas

Después de seleccionar una de las celdas la aplicación mostrará una nueva pantalla la cual contiene información, dentro de una "ScrollView", del tema seleccionado y un botón que nos enviará a una nueva vista con múltiples funcionalidades que se van a mencionar más adelante, además de esto presenta la posibilidad de navegar hacia la pantalla anterior.

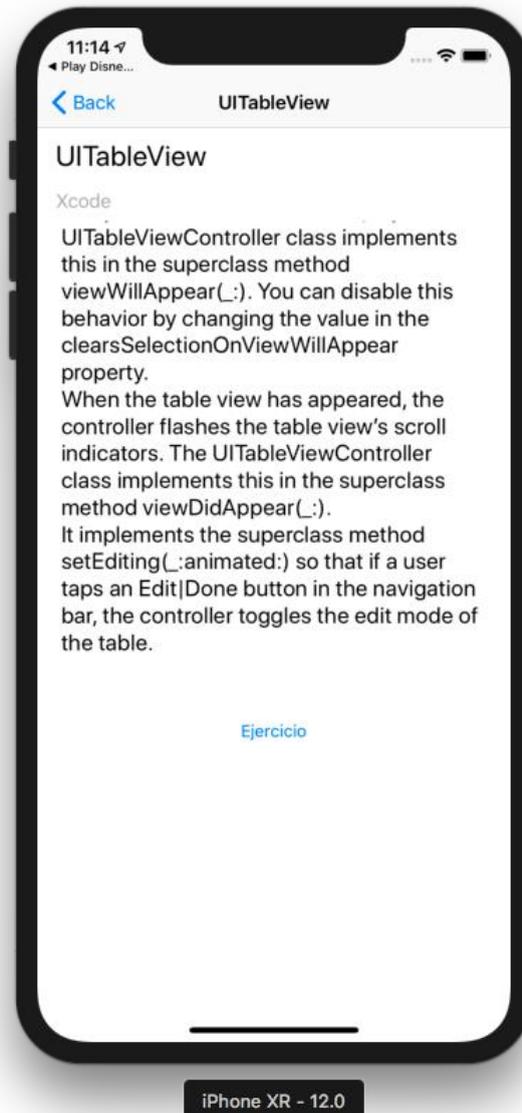
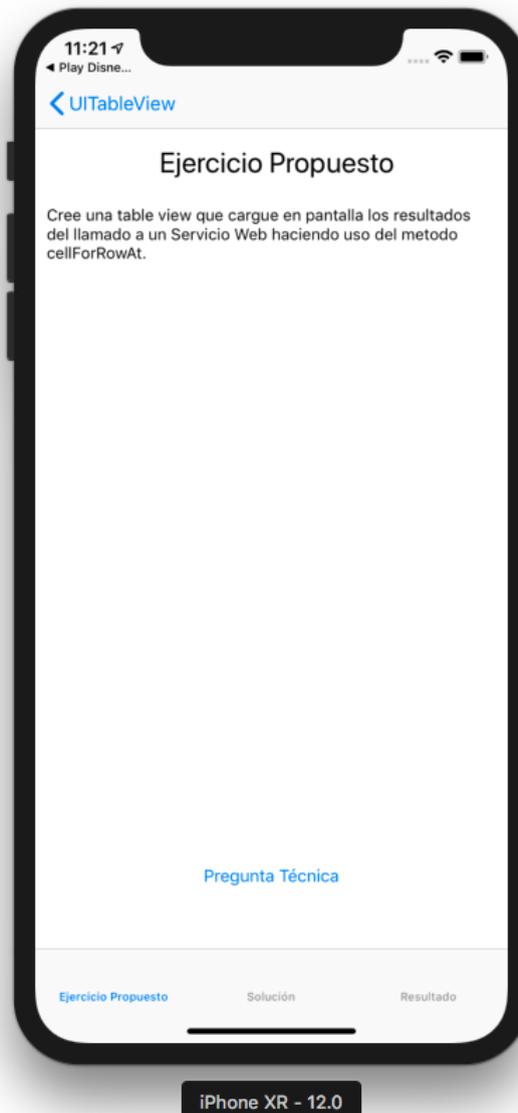


Figura 3. Pantalla con contenido teórico de cada tema.

Cuando se presiona el Botón **Ejercicio** se presenta una nueva vista con múltiples opciones y caminos que puede tomar el usuario, para continuar aprendiendo sobre el tema seleccionado, tal como se muestra en la **Figura 4**.

Figura 4. Pantalla con el ejercicio propuesto y las diferentes opciones.



En esta vista, lo primero que se va a identificar es un ejercicio propuesto, el cual va a variar dependiendo del tema seleccionado y que tiene como objetivo hacer que el estudiante ponga en práctica el concepto en el que está trabajando, la idea de que la pregunta sea tan general es que el estudiante pueda investigar en fuentes externas en caso de que la información brindada por la aplicación no sea suficiente y que además puedan agregar o reducir complejidad dependiendo del nivel actual del usuario.

Justo bajo el ejercicio propuesta se encuentra un Botón que dice **Pregunta Técnica**, el cual enviará al usuario a una pantalla de la cual hablaremos más adelante. Finalmente se puede observar un “TabBar”, que también es una clase proporcionada por el UIKit de Apple y que tiene como objetivo facilitar la navegación entre múltiples vistas dentro de una aplicación, que este caso son 3, la primera vista vista se llama **Ejercicio Propuesto** y es la que mencionamos anteriormente; La segunda vista se llama **Solución** y tiene como objetivo presentar una posible solución (debido a que pueden existir múltiples soluciones) al ejercicio propuesto en la vista anterior, tal como se muestra en la **Figura 5**.

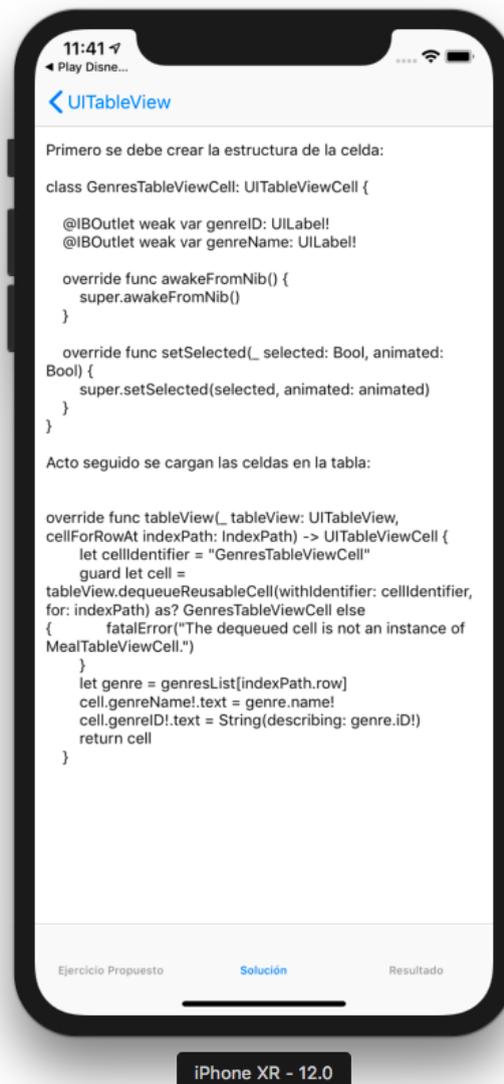


Figura 5. Posible solución al ejercicio propuesto.

Finalmente la última opción disponible se llama **Resultado** y tiene como objetivo mostrar cómo se podría ver una implementación adecuada del concepto que se está trabajando, que en el caso de una TableView se debería ver como se muestra en la **Figura 6**.

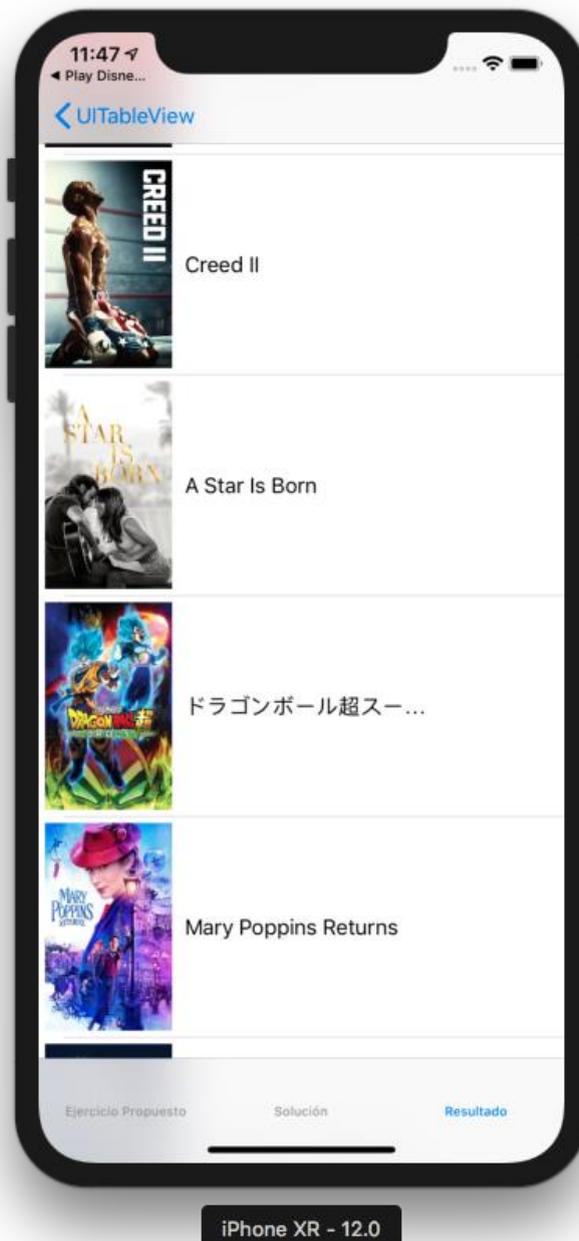


Figura 6. Ejemplo de una TableView.

Para este ejemplo se utilizaron los conceptos sobre TableView mencionados en la aplicación, además de hacer uso de TheMovieDB (<https://www.themoviedb.org/documentation/api>), una API **autorizada por la empresa** que permite consultar diferente información sobre películas y series, en este caso se hizo una consulta sobre las películas más taquilleras actualmente.

Para finalizar regresamos al Botón de **Pregunta Técnica** la cual, como su nombre lo dice, muestra una pregunta técnica con cuatro opciones representadas como botones, los cuales van a desplegar un mensaje en caso de que la respuesta sea correcta o incorrecta.

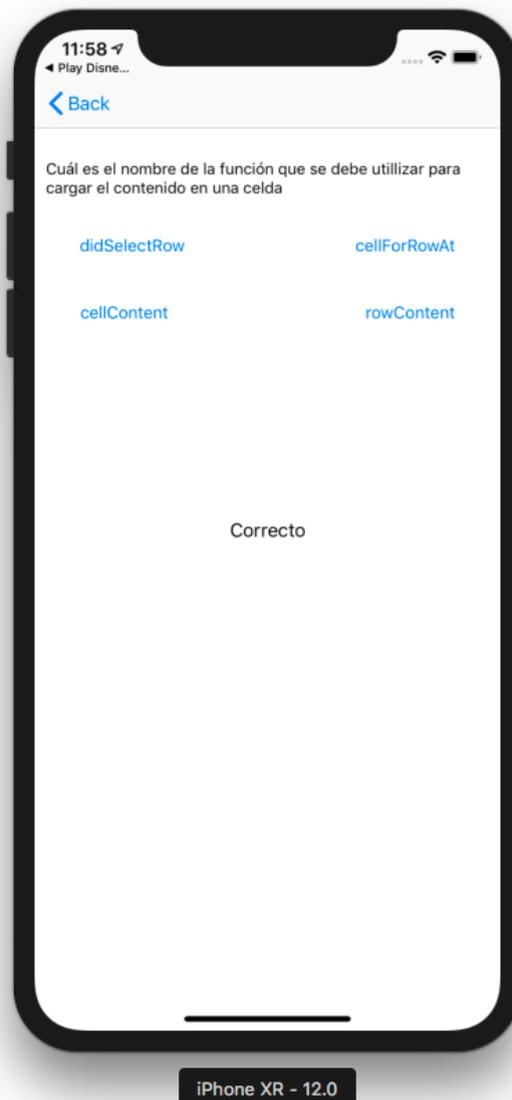


Figura 7. Pregunta Técnica.

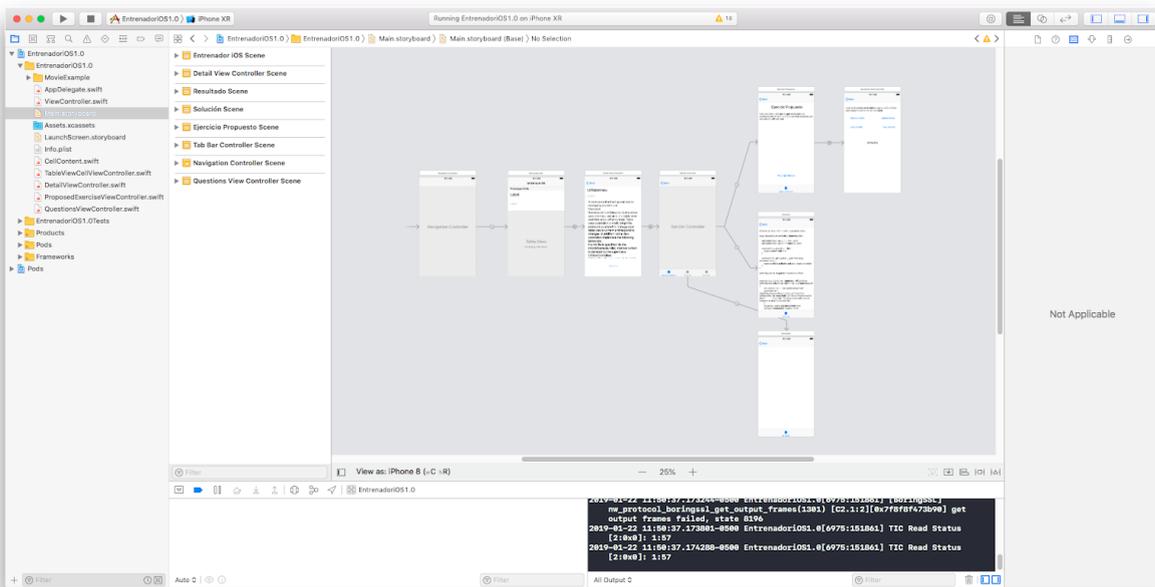
Finalmente la pantalla de pregunta técnica presenta un cuestionario el cual está compuesto por varios Labels que contienen la pregunta y el indicador de si se respondió correcta o incorrectamente, además de esto las cuatro opciones están representadas con botones, los cuales al ser seleccionados van a informar si la respuesta seleccionada fue correcta o incorrecta, es bastante simple pero en caso de que se decida continuar con el proyecto se planean agregar más preguntas y mejorar el diseño del cuestionario.

El objetivo de este cuestionario es identificar si el practicante domina el concepto evaluado, haciendo preguntas muy específicas, que se podrían hacer en una entrevista técnica para un desarrollador iOS.

La evaluación de los conocimientos totales se completa cuando el estudiante responde correctamente cada pregunta dentro de cada tema, por lo tanto en el caso actual serían 19 preguntas, después de esto el practicante debería estar preparado para pasar una prueba técnica básica de iOS (profe esta funcionalidad no está implementada entonces).

Toda esta aplicación se desarrolló haciendo uso de Xcode y el lenguaje de programación Swift, por políticas de la empresa no es posible mostrar capturas del código pero en la **Figura 8** se muestra una imagen del Storyboard, una herramienta de Xcode que permite visualizar la estructura de las vistas, sus conexiones y algo de información extra.

Figura 8. Xcode y el Storyboard.



Toda la información recopilada es mantenida por la empresa, específicamente el Entrenador, para seguir siendo utilizada, en los procesos de capacitación futuros, además de esto se plantea revisar constantemente la información para que esta se mantenga al día respecto a las nuevas tecnologías.

Conclusiones

- El documento realizado y la Prueba de Concepto diseñada fueron evaluadas por el Entrenador de practicantes en el área de iOS y éste consideró que podría ser una aplicación muy útil debido a que ahora el proceso de aprendizaje no dependería únicamente del entrenador (debido a que cuando el entrenador no estaba disponible el proceso se estancaba) y de esta forma los practicantes ahora podrán avanzar en su proceso de forma autónoma.
- Tras el desarrollo de la práctica se pudo alcanzar todos los objetivos específicos propuestos y por ende, el objetivo general. No se tuvieron inconvenientes que afectaron en gran medida el correcto desarrollo del proyecto.
- Cabe recalcar que este aplicativo no fue todo lo que se realizó en las prácticas, y que el trabajo real, no puede ser presentado debido a las políticas de seguridad de Globant y Disney.
- Debido a los inconvenientes presentados debido a las políticas de seguridad de las empresas, se resalta la importancia que empresas del calibre de Disney y Globant, le dan a la propiedad intelectual.
- Aunque el proyecto presentado en este informe no hace parte de lo que se trabajó realmente dentro de la empresa, se hizo uso de todas los conceptos y técnicas aprendidas dentro de esta, por lo cual se puede considerar que todo el proceso de prácticas fue satisfactorio.
- Durante el periodo de tiempo que duró el desarrollo de la aplicación, se pudo comprender varios puntos importantes de los procesos que lleva a cabo Globant en las distintas etapas de un proceso de desarrollo.
- El proyecto cumplió con los objetivos esperados dentro de la empresa y existe la posibilidad de que se continúe desarrollando dentro de esta.
- Durante el desarrollo del proyecto y el estudio del problema, se logró adquirir mucho conocimiento de negocio respecto a los distintos procesos que se manejan en la empresa Globant.
- El de entrenamiento me permitió aprender mucho sobre diferentes temas que antes eran desconocidos para mí (Swift, Xcode, Design Patterns y Scrum).

Referencias bibliográficas

- [1]"Home Page | Globant", Globant.com, 2018. [Online]. Available: <https://www.globant.com/>. [Accessed: 15-Sep-2018].
- [2]"iOS 12", Apple (Latin America), 2018. [Online]. Available: <https://www.apple.com/lae/ios/ios-12/>. [Accessed: 15-Sep-2018].
- [3]"The Importance of Training and Development in the Workplace", 2020 Project Management, 2018. [Online]. Available: <http://2020projectmanagement.com/resources/project-management-training-and-qualifications/the-importance-of-training-and-development-in-the-workplace>. [Accessed: 15-Sep-2018].
- [4]"Human Interface Guidelines - Design - Apple Developer", Developer.apple.com, 2018. [Online]. Available: <https://developer.apple.com/design/human-interface-guidelines/>. [Accessed: 15-Sep-2018].
- [5]"Xcode - Apple Developer", Developer.apple.com, 2018. [Online]. Available: <https://developer.apple.com/xcode/>. [Accessed: 15-Sep-2018].
- [6]"Swift - Apple Developer", Developer.apple.com, 2018. [Online]. Available: <https://developer.apple.com/swift/>. [Accessed: 15-Sep-2018].
- [7]"Alamofire/Alamofire", GitHub, 2018. [Online]. Available: <https://github.com/Alamofire/Alamofire>. [Accessed: 15-Sep-2018].
- [8]C. Team, "CocoaPods.org", Cocoapods.org, 2018. [Online]. Available: <https://cocoapods.org/>. [Accessed: 15-Sep-2018].
- [9]"Qué es SCRUM", Proyectos Ágiles, 2018. [Online]. Available: <https://proyectosagiles.org/que-es-scrum/>. [Accessed: 15-Sep-2018].