Actividad 3 (Individual o Grupal)

Desarrollo de una aplicación móvil multiplataforma

**Si esta actividad se realiza en grupo, la composición del grupo debe quedar registrada en la sección de equipos de esta actividad. Dicha sección será visible unos días antes de la entrega de la actividad. No registrar el grupo podría conllevar la no evaluación de la actividad.**

El objetivo de las tres actividades de la asignatura *Aplicaciones Móviles Multiplataforma* es que seas capaz de desarrollar varias aplicaciones multiplataforma que se ejecuten en distintos tipos de dispositivos utilizando un servidor web simulado (Back-End simulado). Es decir, abarcaremos únicamente el diseño y el desarrollo de la interfaz que el usuario maneja (desde distintos dispositivos).

Una vez terminadas, habrás sido capaz de crear una API simulada con datos de prueba para ayudarte en tus desarrollos y habrás integrado aplicaciones web y aplicaciones móviles multiplataforma desarrolladas con React y React Native contra dicha API.

Imagen de la pantalla de un celular con la imagen de una caricatura

El contenido generado por IA puede ser incorrecto.

Figura - Objetivo final de las actividades de la asignatura

En la actividad anterior se desarrolló una aplicación web SPA haciendo uso de la librería React y otros módulos externos como React Router. **El objetivo de esta tercera y última actividad es desarrollar una aplicación móvil multiplataforma haciendo uso de React Native que haga exactamente lo mismo que hace la versión web, pero de forma nativa para cada una de las plataformas móviles más usadas hoy en día (Android y iOS)**. Para ello haremos uso de React Native y de los componentes y técnicas estudiados en la asignatura:

* **SafeAreaView, View, Text, Pressable**… Será necesario demostrar un uso coherente y adecuado de los diferentes componentes que se incluyen en la distribución de React Native y Expo.
* **Uso de retroalimentación háptica**. Al menos una sección de la aplicación debe ofrecer retroalimentación háptica al usuario. Queda a elección del estudiante la ubicación en la que implementar este comportamiento. Se valorará dicha ubicación y la retroalimentación elegida.
* **Ejecución en dispositivos** **virtuales y físicos**. Se deberá demostrar que la aplicación funciona correctamente al menos en un dispositivo virtual.
* **Estilizado con Nativewind**. La aplicación escrita usando React Native debe usar Nativewind para el estilizado. Se permitirá aun así usar StyleSheet de React Native cuando no sea técnicamente posible usar Nativewind. Obligatoriamente se deberán incluir al menos dos fuentes (que se hayan descargado) a la aplicación.
* **Navegación mediante Stack, Tabs o Drawer**. Al menos dos de los tres tipos de navegación estudiados deben implementarse en la aplicación. Queda a elección del estudiante cuáles usar.

En la actividad anterior se desarrollaron (o se debieron desarrollar) las siguientes vistas:

* Vista inicial o landing. Obligatoriamente debe ser implementada.
* Otras cinco vistas a elección. Obligatoriamente deben ser implementadas, aunque no es necesario que sean exactamente esas. Es decir, pueden ser otras nuevas siempre que sean en total cinco vistas (más la landing, es decir, 6 en total).
* Una vista de *login* y al menos una ruta protegida. No es obligatorio replicar esto en nuestra aplicación multiplataforma nativa para reducir la complejidad de la actividad, pero será necesario para obtener la máxima calificación.

Al igual que ocurría en la versión web, **deben usarse los datos incluidos en la API simulada desarrollada en la primera actividad de la asignatura.** Por otro lado, se sobreentiende que una gran parte del código de la actividad anterior puede ser reutilizado realizando las debidas modificaciones para adaptarlo a React Native. Algunos otros componentes como los hooks podrán reutilizarse, seguramente, de forma completa. De esta forma evidenciamos una de las ventajas de usar React Native como tecnología para aplicaciones móviles y React como tecnología para la web.

Sírvete del [código de apoyo de la asignatura que encontrarás en GitHub](https://github.com/UnirCs/DM-GLOBAL) para poder tener algo de ayuda a la hora de realizar esta actividad.

Para documentar todo el proceso realizado será obligatorio realizar una vídeo memoria donde todos los integrantes del equipo, en el caso de realizarse la actividad de manera grupal, deberán participar usando tiempos similares. En caso de realizarse la actividad de forma individual también es necesario realizarla.

La vídeo memoria tendrá una duración máxima aproximada de 15 minutos y deberá visitar los siguientes aspectos de tu actividad:

1. **Navegación.** Se debe indicar las rutas que tiene la aplicación y como está configurada.
2. **Vistas.** Se deben explicar las diferentes vistas (su código) y como están formadas (composición de componentes, si hay retroalimentación háptica, uso de hooks...)
3. **Tour virtual.** Se debe realizar un tour virtual por todas las vistas y rutas de la aplicación. Además, se mostrará que las peticiones que se realizan para obtener la información de las diferentes vistas son contra la API simulada.
4. **Conclusiones.** Se añadirá cualquier comentario que se desee, así como feedback.

La entrega consistirá **un único archivo ZIP** con el contenido del proyecto desarrollado, es decir:

* **Archivo** Autores.txt: Contiene los nombres completos de los autores del trabajo. Únicamente en caso de que se entregue en grupo.
* **Proyecto con el código de la aplicación hecha con React Native** sin incluir la carpeta node\_modules y sin comprimir.
* **Vídeo memoria** **obligatoria** en formato MP4 de la actividad. En caso de que la vídeo memoria pese demasiado y no pueda adjuntarse en la entrega del Aula Virtual deberá incluirse un enlace a la misma subida al OneDrive de la cuenta de UNIR, asegurando que el enlace es público (es fácil comprobar esto tratando de acceder al enlace desde una pestaña de incógnito del navegador en la que no se haya iniciado sesión en OneDrive).

Rúbrica

|  |  |  |  |
| --- | --- | --- | --- |
| Criterio | Descripción | Puntuación máxima  (puntos) | Peso  % |
| Criterio 1 | Se implementa la landing de la aplicación | 0,5 | 5 % |
| Criterio 2 | Implementadas cinco vistas a elección en función de la temática de la aplicación. | 2,5 | 25 % |
| Criterio 3 | Se usan al menos dos tipos de navegación y la navegabilidad es correcta | 2 | 20 % |
| Criterio 4 | Se utiliza Nativewind como librería para la gestión de estilos. Inclusión de dos fuentes distintas a las que vienen por defecto en la aplicación | 2 | 20 % |
| Criterio 5 | Se ofrece retroalimentación háptica en al menos un punto de la aplicación y dicho lugar y retroalimentación son coherentes | 1 | 10 % |
| Criterio 6 | La aplicación usa los datos recibidos de una API simulada realizada previamente | 1 | 10 % |
| Criterio 7 | **Elaboración de la vídeo memoria** **obligatoria** según las especificaciones, ajustándose a la duración aproximada y cubriendo los aspectos solicitados. | 1 | 10 % |
|  |  | **10** | **100 %** |