



Práctica

Visión General

A través de Eh-Ho hemos aprendido gran parte de los fundamentos de Android. Es hora de crecer nuestra aplicación y aplicar tus conocimientos dentro de la plataforma.

Objetivos

1. **Aplicación de los conocimientos de la plataforma:** Uso de los componentes principales de Android
2. **Aplicación de los conocimientos de Kotlin:** Uso de los conceptos de programación orientada a objetos y programación funcional
3. **Diseño de la aplicación:** Seguir las guidelines de Material Design para la construcción de la interfaz



Especificaciones

Lo que haremos será añadir una serie de funcionalidades a la aplicación de Eh-Ho, para ello se proveerá una historia de usuario y una propuesta para su desarrollo

1. Validación de formularios

Historia de usuario

Los diferentes campos de SignIn/SignUp deber ser validados y, en caso de no ser correctos, notificar a los usuarios acerca de ello.

Detalles de Implementación

- Se deberá mostrar una etiqueta de error en los EditText
- Email debe ser un email válido
- Username debe ser una cadena de texto mayor de 5 caracteres alfanuméricos
- Password debe ser una cadena de texto mayor de 8 caracteres que contenga mayúsculas, minúsculas, números y al menos un símbolo



2. Manejo de errores

Historia de usuario

En la aplicación sólo se ha tenido en cuenta el Happy-Path, ignorando el caso en el que se produzca un error, ya sea por la conexión de red u otro caso. El usuario debe ser notificado de cualquier tipo de error que suceda en la aplicación que sea relevante para él

Detalles de implementación

- Los errores deben ser mostrados al usuario sin bloquearle su interacción a menos que sea requerida una acción por parte del usuario

3. Loading/Empty states

Historia de usuario

El usuario debe tener constancia en todo momento de qué está pasando en la aplicación y que no sienta que la aplicación se ha quedado colgada si una operación en segundo plano de carga de datos se demora demasiado por diversas causas. Si por algún caso el listado solicitado por el usuario es vacío, el usuario debe visualizar un empty state indicándole tal resultado

Detalles de Implementación

- Durante la carga de datos se deberá mostrar un ProgressBar indicando al usuario que se está realizando una carga de datos.
- En el caso de no recibir ningún contenido, se deberá mostrar al usuario que no se han recibido ningún dato

4. Pull-To-Refresh

Historia de Usuario

El usuario debe ser capaz de refrescar el listado por medio del gesto Pull-To-Refresh

Detalles de implementación

- Uso de Pull-To-Refresh en el RecyclerView



5. Mejorar la UI del listado de topics

Historia de usuario

El listado de topics debe mostrar más información al usuario sin necesidad de acceder dentro del detalle. Para ello se modificará la UI de cada fila incluyendo los siguientes datos:

- Usuario del último comentario
- Número de respuestas
- Número de Likes
- Cualquier otra información que se vea relevante para el usuario

Además de los campos anteriores, el listado de topics debe renderizar diferentes vistas dependiendo de si el mensaje ha sido “pineado”, “bumpado” o es un topic normal

Detalles de implementación

- El modelo de datos y el ViewHolder debe ser actualizado con los nuevos datos
- Sería interesante tener diferentes ViewHolders por cada tipo de fila

6. Listado de posts

Historia de usuario

El usuario debe ser capaz de acceder al listado de posts de cada uno de los topics

Detalles de implementación

- Crear una nueva pantalla dónde se muestren el listado de post
- A ésta nueva pantalla se accederá al hacer click sobre un topics
- Cada post debe mostrar la información relevante

7. Creación de Post

Historia de usuario

El usuario debe ser capaz de crear un nuevo post a un topic

Detalles de implementación

- Se deberá crear un formulario dónde el usuario pueda rellenar la información del post
- Revisar documentación del endpoint:

<https://docs.discourse.org/#tag/Topics/paths/~1posts.json/post>



Evaluación

Se deberán entregar las 7 tareas anteriores implementadas sobre la aplicación base que se provee al alumno. El alumno puede añadir cualquier otra mejora que considere oportuna. Se valorará la calidad del código fuente así como el uso de Kotlin y la aplicación del patrón de diseño MVVM. Se valorará positivamente la creación de tests que comprueben el correcto funcionamiento de las partes más delicadas de la aplicación