

Mitología Ibérica App, Instagram flutter clone.

Documentación Técnica.

En este documento comentaré algunas de las decisiones de diseño que he tomado para la aplicación, así como problemas que han podido surgir durante el desarrollo para poder arreglarlo en caso de ocurrir a otro usuario o a mí mismo en un futuro.

Lo primero que quiero comentar es que para programar la aplicación se ha utilizado el emulador "Pixel 3ª API 33 x86 64 (mobile)". En caso de usar un emulador diferente es posible que cambie visualmente el programa debido a los tamaños de pantalla usable por lo que se recomienda usar este emulador.

Para empezar con los problemas ocurridos y como solucionarlos voy a comentar el que más veces me ha ocurrido. El desarrollo se ha realizado en dos equipos diferentes y esto, en ocasiones ha significado encontrar problemas en el proyecto al pasar de uno a otro.

Si el proyecto no compila al abrirlo hacer los siguientes pasos:

- 1- En caso de no coincidir la versión del proyecto con la versión de flutter soportada por el IDE utilizado hay dos opciones:
 - a. upgradear la versión del sdk.
 - b. cambiar la versión del proyecto por si es compatible. Para ello nos dirigimos al archivo "pubsec.yaml", buscamos donde pone "enviroment" y cambiamos la versión del sdk a la esperada por nuestro IDE.

Por ejemplo, si encontramos:

```
environment:  
  sdk: '>=2.18.4 <3.0.0'
```

y nuestro IDE espera la versión 2.18.2 podemos poner:

```
environment:  
  sdk: '>=2.18.2 <3.0.0'
```

- 2- Abrir terminal y poner los siguientes comandos:

- flutter clean
- flutter packages get
- flutter packages upgrade

- 3- Restart Android Studio (File > Restart IDE > Acept)

Otro de los problemas que he encontrado es referente a la versión del SDK de Android, aunque es un 'warning' y no un error como tal. Algunos de los plugins utilizados como, por ejemplo, el encargado de abrir enlaces de forma correcta sugería actualizar la SDK de Android para subirla a la 33. Es por ello por lo que voy a explicar cómo actualizarlo por ocurre lo mismo:

- 1- Para actualizar el SDK de Android en Android Studio es necesario ir a Tools > SDK Manager.
- 2- Dentro de esta ventana aparecen tres pestañas, vamos primero a SDK Platforms, que debería ser la que se abre por defecto. Allí Seleccionamos Android Tiramisu Preview.
- 3- Luego vamos a la pestaña SDK Tools y seleccionamos Android SDK Build-Tools 33 (puede darse el caso de que no aparezca el número y solo salga Android SDK Build-Tools).
- 4- Por último, hacemos click en Ok y esperamos que se actualice.

Durante el desarrollo me di cuenta de que la aplicación podía emularse directamente sobre mi dispositivo móvil sin usar el emulador de Android Studio, lo probé y al hacerlo me di cuenta de que la aplicación se instalaba en mi dispositivo, pero el icono de esta era el de flutter así que busqué si era posible cambiarlo, aquí explico la forma que he encontrado que me ha funcionado a mí.

Para cambiar el logo de la aplicación:

- 1- Crear directorio assets/icon/ y guardar dentro el icono a utilizar llamado icon.png (Se recomienda que tenga la mayor resolución posible para que no se vea pixelado en dispositivos de alta densidad de pixeles)

Nota: No es obligatorio ese nombre ni ese directorio, pero al poner otros diferentes puede causar problemas dependiendo la versión de la dependencia por lo que es mejor hacerlo así para evitar errores.

- 2- En el archivo pubsec.yaml (otro si la configuración del proyecto tiene otro nombre en tu caso) buscamos donde ponga dev_dependencies y añadimos la siguiente:

```
dev_dependencies:  
  flutter_launcher_icons: "^0.11.0"
```

- 3- Esta dependencia creada por la comunidad nos importa unos paquetes para hacer posible el cambio de icono. Ahora debemos añadir lo siguiente

```
flutter_icons:  
  android: "launcher_icon"  
  ios: true  
  image_path: "assets/icon/icon.png"  
  min_sdk_android: 21
```

Estas líneas serán las encargadas de decirle al plugin que icono queremos poner y algunas opciones más como por ejemplo que funcione en IOS, la SDK mínima que debe tener Android para usar nuestro icono (se recomienda la 21) y si en Android queremos sustituirlo por el anterior icono (flutter) o dejar los dos. Esto último es por si el nuevo icono falla, que no se quede sin icono, pero indicando que tiene prioridad al nuestro (es el caso utilizado).

Ahora bien, en cuanto a diseño de la aplicación quiero comentar algunas de las decisiones tomadas.

El login de la aplicación es totalmente inútil e innecesario, pero me tomo el desarrollo de esta aplicación como una manera de aprender cosas nuevas en flutter y por eso ha sido implementado a pesar de no haber registro de usuarios ni validación de los campos introducidos.

En cuanto a la navegación por la aplicación me parecería mucho más intuitiva hacerla con una barra inferior de navegación por iconos en lugar de un drawer desplegable, sin embargo, era una característica necesaria de la aplicación que existiese el drawer y poner ambas maneras me parecía demasiado recargado e innecesario. Por este motivo la navegación se ha realizado con un drawer.

En la pantalla del perfil tenía dos opciones para hacerla deslizable hacia abajo, o bien hacerlo de la pantalla completa o bien hacerlo solo de la parte de las publicaciones. He tomado la decisión de usar la segunda y explico por qué:

- 1- La primera razón es que en la pantalla de Home ya utilizo un deslizable de pantalla completa y esto era una forma de utilizar otra diferente.
- 2- En el caso concreto de mi aplicación, se ven suficientes publicaciones ya que la biografía y las historias no ocupan demasiado espacio (restricción heredada de la imagen a clonar) y por ello creo que es más útil deslizar solo el apartado de publicaciones y tener el resto de información del perfil visible en todo momento.

Antes de acabar quiero comentar la refactorización de código llevada en la aplicación. El desarrollo de esta aplicación la comencé sin mucha idea de programar en flutter y del lenguaje Dart, tenía los conocimientos más básicos y por tanto los códigos eran menos limpios e intuitivos. Según he avanzado en el desarrollo he aprendido mejores formas de hacer algunas cosas o simplemente a refactorizar de una mejor manera.

Es posible que no sea la mejor manera de organizarlo, pero creo que es mucho mejor que la que usaba al principio del desarrollo así que estoy contento con esa evolución.

Por último, quiero comentar una decisión que tiene que ver con el estilo visual. Se trata de los enlaces.

En la aplicación encontramos diferentes enlaces, uno de ellos al manual de uso de la aplicación, otro a la página web de igual temática realizada para la asignatura de Lenguaje de Marcas un año antes del desarrollo de esta aplicación y algunos más.

Estos enlaces aparecen de color azul y subrayados y en ocasiones puede parecer que difiera del estilo del resto de fuentes, tamaños y colores de la aplicación. Esto he decidido que quede así para un mayor entendimiento del usuario de que esos enlaces son seleccionables para navegar y usar el mismo estilo visual en todos los enlaces de la aplicación para no despistar al usuario de cuales son seleccionables y cuáles no.