

World Capitals App Documentación

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 y funcionalidades que me habría gustado implementar pero que no ha sido posible en el tiempo estimado de la tarea.

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 (aunque se ha tratado de que sea lo más responsive posible) debido a los tamaños de pantalla usable por lo que se recomienda usar este emulador.

Para empezar con los problemas ocurridos y cómo 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 > Accept

Para cambiar el logo de la aplicación:

1- Crear directorio assets/icon/

2- 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 píxeles)

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.

3- 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"
```

4- 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.

La aplicación consiste en un método para practicar y estudiar las capitales de los diferentes países del mundo de una forma más interactiva.

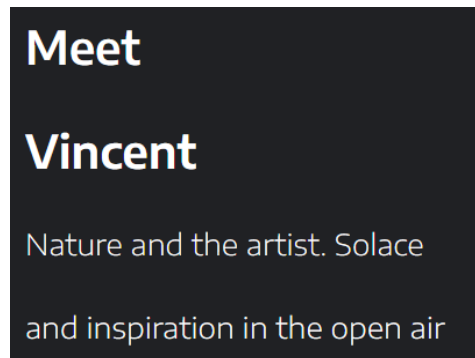
En cuanto a paleta de colores me he inspirado en la paleta de colores usada en el museo de Van Gogh de Amsterdam ya que usa colores inspirados en la naturaleza debido a los cuadros en los que se basa.



Verdes de los campos, amarillos del sol y azules de océanos y cielos. Buscaba una paleta que se acercara a estos colores debido a la temática de la aplicación y finalmente encontré esta que además de cumplir lo que buscaba me parece bastante estética para la aplicación.



Para la tipografía, también me he inspirado en la misma fuente redondeada del museo y he encontrado una similar de google fonts que me ha parecido bastante estética e interesante. **ENCODE SANS**.



En cuanto a **funcionalidades**, en la aplicación encontramos lo siguiente:

- Tiene un registro de usuarios tanto por email como por cuenta de google.
- Tiene inicio de sesión de los usuarios por email y por google.
- Tiene una opción para resetear la contraseña en caso de olvidarla.
- Una opción es estudiar las capitales.
- Otra opción es jugar partidas en las que se muestra el nombre de un país y el usuario debe acertar la capital correcta entre cuatro opciones.
- Por último tenemos la opción de ver las mejores puntuaciones obtenidas.

Había más ideas para ampliar esta aplicación que comentaré posteriormente pero que no han sido llevadas a cabo por falta de tiempo y sobre todo por intentar mejorar el código.

Buscando cómo implementar el registro de usuarios en firebase debido a un fallo que me surgió encontré un curso que además de explicar esto que me hacía falta explicaba de manera muy buena como seguir buenas prácticas de código limpio.

En este punto tomé la decisión de rehacer buena parte del código siguiendo estas prácticas en lugar de implementar funciones nuevas ya que creo que es mucho más sencillo ampliarlo una vez se tenga un mejor código y sobre todo ya que era algo nuevo que aprender sobre flutter de cara a próximos proyectos.

Esto ha llevado como he mencionado a que algunas funciones se queden sin implementar pero que estaban previstas y son las siguientes:

- Lo primero que quería comentar es respecto al json utilizado de países-capitales. Tiene algunos “fallos” que me gustaría corregir en el futuro. Por ejemplo, aparecen países que no tienen capital y en lugar de aparecer

“No tiene” aparece vacío. También ocurre en ocasiones que aparecen territorios como Hong Kong que no es realmente un país.

Estos fallos no me parecen muy graves ya que quiero rehacer el json añadiendo las banderas (comentaré más adelante) y de paso arreglando estos pequeños fallos encontrados. He probado con bastantes json y apis de internet y en todos he encontrado este tipo de incoherencias por lo que he optado por dejarlo así y centrar mi esfuerzo en otras funciones.

- Un selector de dificultad para las partidas que ponga más opciones para responder, que haga las partidas más largas o incluso con tiempo para responder cada pregunta.
- Otra idea era que además de poder practicar las capitales se pudiesen practicar las banderas, de hecho sería bastante sencillo implementarlo ya que se pueden cargar las imágenes de las banderas con la url desde el mismo json y no haría falta descargarlas todas dentro del proyecto.
- Obviamente esto también sería llevado a la parte de estudio para poder estudiar las banderas.
- Otra función a implementar es en cuanto a las puntuaciones, diferenciando entre puntuaciones propias y las globales de la aplicación (que son las únicas visibles actualmente), o incluso un ranking con tus amigos (aunque esto último era más inviable en el plazo dado)
- También respecto a las puntuaciones me gustaría añadir que el usuario obtuviese puntos extra por racha seguida de aciertos para que haya más variedad de puntuaciones y no solo sean en función de cuántas se aciertan y se fallan.
- Por último y como funcionalidad menor, me gustaría cambiar la selección de respuestas durante una partida.
En la versión actual, en el momento de hacer click en una respuesta, aparece la nueva pregunta automáticamente (finaliza el juego en caso de ser la última).

Me gustaría que en la nueva versión, al seleccionar una respuesta, te indique si has acertado o fallado cambiando el color de fondo de la respuesta a verde o rojo respectivamente y en ese momento aparezca un botón de continuar en la parte inferior para ir a la siguiente pregunta. De esta manera sería más intuitivo para el usuario.

Obviamente habría más mejoras que me gustaría añadir como sonidos a la aplicación, cambiar entre tema oscuro y claro, un apartado de ajustes o la posibilidad de tener amigos dentro de la app pero esto son funcionalidades menos importantes para el funcionamiento básico de la aplicación que no se han priorizado.

Otros datos del desarrollo:

En la pantalla principal podemos observar una muestra de lo que podría ser un modo oscuro y en el resto de ventanas se ha usado un tema más claro. Me hubiera gustado tener todas las ventanas con los dos modos y poder cambiarlo pero no ha dado tiempo.

También en la pantalla principal podemos encontrar un gif de una bola del mundo que en mi opinión queda muy estético y le da un toque diferencial a la aplicación.

Sin embargo, esto hace que el emulador se ralentice más de lo que debería por lo que también he buscado una opción que lo sustituya por un icono del mundo estático. Ambas funciones se pueden encontrar en el archivo 'home_page.dart'.

Esto en un principio no debería ser problema en dispositivos móviles reales (en mi caso funciona perfectamente el gif) pero se podría activar una opción en el futuro en ajustes, por si algún usuario tiene problemas de rendimiento para sustituir el gif por el icono y que sea más liviana la aplicación.

También se han usado diferentes cuadros de diálogo, unos más similares a como suelen aparecer en ios y otros más parecidos a lo que solemos encontrar en android. De esta manera se muestra como serían para cada uno de estos sistemas operativos en caso de exportar la aplicación a ambas plataformas.

Ejemplo de cuadro de diálogo utilizado habitualmente en Android:

- Al intentar salir de una partida un cuadro de diálogo advierte de que se perderán los datos de la partida para que el usuario confirme si desea salir.

Ejemplo de cuadro de diálogo utilizado habitualmente en IOS:

- Al introducir mal los campos de Login, por ejemplo usando un usuario que aún no se ha registrado en la aplicación.