**Sesión 3-08 Clase del 5 de febrero**

En esta sesión vamos a programar el acceso a ficheros almacenados en memoria interna. Con memoria externa (SDCard) no vamos a trabajar. Sería muy similar a lo que vamos a programar, aunque se necesitarían agregar permisos a acceso a memoria externa.

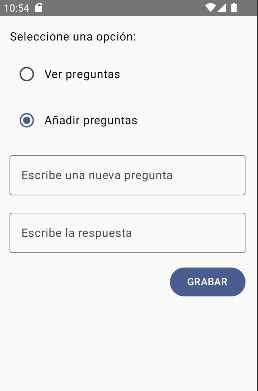
Nuestra app usa un fichero csv donde se pueden añadir preguntas y respuestas. También se pueden ver las preguntas almacenadas en el fichero.

Inicialmente, copia el código del composable correspondiente a la única pantalla que va a tener la app. Lo tienes dentro del archivo **Archivos\_Sesion3\_08.rar** de la carpeta RECURSOS de Moodle

En la pantalla tienes dos RadioButtons. Según el que se seleccione, se oculta/muestra una sección de la pantalla.

Cuando se selecciona **Ver Preguntas,** se tiene una pantalla **destinada a mostrar una pregunta** de las almacenadas en el fichero CSV, con un botón SIGUIENTE destinado a que muestre una nueva pregunta.

Cuando se selecciona **Añadir Preguntas,** se tiene una pantalla **destinada a añadir al fichero CSV el texto de una pregunta y respuesta** editadas en dos **TextFields.** de las almacenadas en el fichero CSV, con un botón SIGUIENTE destinado a que muestre una nueva pregunta.

 Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

En el código tienes ya programada la funcionalidad de los **RadioButtons**, así como variables de estado para el contenido de los **TextFields** y para controlar el **RadioButton** seleccionado.

1.- Vamos a añadir funcionalidad para añadir preguntas y respuestas a un archivo CSV. La siguiente función permite agregar una pregunta con su respuesta al fichero usando un flujo **FileOutputStream.** El código está comentado para facilitar su comprensión.

// Función para guardar en un archivo CSV  
fun saveToCsvFile(context: Context, question: String, answer: String): Boolean {  
 return try {  
 val fileName = "preguntas\_respuestas.csv"  
 //el archivo se crea en el directorio de archivos de la aplicación  
 // en /data/data/com.example.sesion3\_08/files  
 val file = File(context.*filesDir*, fileName)  
 if (!file.exists()) {  
 file.createNewFile()  
 }  
 // abrir el archivo en modo de escritura (añadir al final)  
 val fos = FileOutputStream(file, true)  
 // linea csv con coma separando pregunta y respuesta  
 val csvLine = "\"$question\",\"$answer\"\n"  
 //grabar, cerrar y devolver true si se grabo  
 fos.write(csvLine.*toByteArray*())  
 fos.close()  
 true  
 } catch (e: Exception) {  
 e.printStackTrace()  
 false  
 }  
}

Programa el botón **GRABAR** para que llame a la función **saveToCsvFile** pasándole el contexto y los textos de pregunta y respuesta editados en los TextFields (siempre que no están vacíos).

Si se ha añadido o no la pregunta correctamente se debe escribir un Log con un mensaje que lo indique.

Comprueba la funcionalidad de la app.

2.- Elimina los Logs anteriores para que, en su lugar, se escriban **Toast** indicando el resultado de añadir la pregunta.

3.-A continuación, vamos a programar que en la sección **Ver Preguntas**, cuando se clica **SIGUIENTE** se muestre una pregunta aleatoria. Antes de ello, debes añadir este método que devuelve una lista de las líneas del fichero CSV. Cada línea del fichero CSV se descompone es un array obtenido con **Split.**

// Función para leer preguntas y respuestas desde el archivo CSV

//y devolverlas en una lista de arrays.

// Cada array tiene la pregunta y la respuesta

fun readFromCsvFile(context: Context): List<Array<String>> {  
 val fileName = "preguntas\_respuestas.csv"  
 val file = File(context.*filesDir*, fileName)  
 if (!file.exists()) {  
 return *emptyList*()  
 }  
 return try {  
 return file.*readLines*().*map* **{** line **->** // Dividir la línea en columnas por las comas  
 // y quitar las comillas (si las hubiera) que encierran a pregunta o respuesta  
 line.*split*(",").*map* **{ it**.*trim*('"') **}**.*toTypedArray*()  
 **}** } catch (e: Exception) {  
 e.printStackTrace()  
 *emptyList*()  
 }  
}

Declara dos variables de estado destinadas a contener el texto de una pregunta y el texto de una respuesta.

Crea una función que recibe un objeto **List<Array<String>>** del contenido del fichero CSV y asigna a las variables de estado anteriores el contenido de una pregunta y su respuesta obtenidas aleatoriamente de la lista.

Para obtener una línea aleatoria de la lista, puedes usar este código:

val data = *readFromCsvFile*(context)  
if (data.*isNotEmpty*()) {  
 val randomLinea = data[Random.nextInt(data.size)]

}

4.- Programa que al clicar el botón **SIGUIENTE** se llame a la función creada anteriormente y que en el componente **Text** destinado al contenido de la pregunta se muestre ese contenido (la variable de estado para la pregunta).

Prueba la funcionalidad.

5.- Establece que, cuando se inicia por primera vez la pantalla, se muestre una pregunta aleatoria.

6.- Añade al botón SIGUIENTE funcionalidad para que, al clicarlo, antes de mostrar el texto de la pregunta siguiente, se muestre en un Toast la respuesta a la pregunta anterior que estaba escrita en el componente Text.