

A dark blue vertical bar is positioned on the left side of the page. A blue arrow-shaped banner points to the right from this bar, containing the date '10-10-2024'. In the bottom-left corner, there are several thin, curved lines in shades of blue and grey, resembling stylized grass or reeds.

10-10-2024

Práctica 6. CodeLab: Persistencia de datos

Acorán González Moray

Contenido

1. Enlace 2

2. Captura de Pantalla..... 2

3. Opinion..... 5

1. Enlace

Aquí señalo **2 enlaces**, el primero hace referencia al **trabajo final de asignatura** 'QRStockMate' como usted habia comentado, dado que ya tenemos implementado el backend con sus respectivas bases de datos (**sqlserver, firebasestorage**) nos comento que con pasarle ese enlace bastaba, pero que hicieramos una pequeña prueba con datastore, este ultimo sería el 2 enlace

Recuerde que se le hizo colaborador al usuario **ivanhdezfragiel** dado que el repositorio esta en privado, por tanto, debería entrar mediante ese usuario para que pueda visualizar el proyecto.

1- <https://github.com/AcoranGonzalezMoray/QRStockMate>

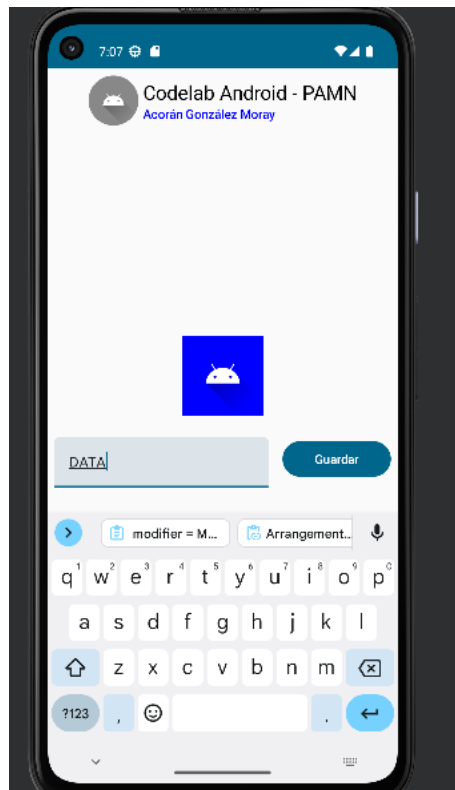
2- <https://github.com/AcoranGonzalezMoray/AplicacionesMovilesNativas/tree/main>

2. Captura de Pantalla

GUARDANDO TEXTO EN DATASTORE:



MODIFICANDO TEXTO SIN GUARDAR:



CERRANDO APP PARA VOLVER A ENTRAR Y RECUPERAR VALOR GUARDADO:



Funciones principales:

- Repository

```
class SettingsRepository(private val datastore: DataStore<Preferences>) {

    suspend fun saveValues(name:String, checked:Boolean){
        datastore.edit { preference ->
            preference[stringPreferencesKey( name: "name")] = name
            preference[booleanPreferencesKey( name: "vip")] = checked
        }
    }

    fun getValues()= datastore.data.map { preference ->
        UserProfile(
            name = preference[stringPreferencesKey( name: "name")].orEmpty(),
            vip = preference[booleanPreferencesKey( name: "vip")]?:false
        )
    }
}
```

- Llamada desde Activity, jugando con hilos para no afectar al hilo main:

```
fun SaveData(textValue: String) {
    val settings = SettingsRepository(dataStore = datastore)

    lifecycleScope.launch(Dispatchers.IO){ this: CoroutineScope
        settings.saveValues(textValue, checked: true)
    }
}

suspend fun LoadData(): String {
    val settings = SettingsRepository(dataStore = datastore)
    val nameDeferred = CompletableDeferred<String>()

    lifecycleScope.launch(Dispatchers.IO){ this: CoroutineScope
        settings.getValues().collect{ it: UserProfile
            withContext(Dispatchers.Main){ this: CoroutineScope
                val name = it.name
                nameDeferred.complete(name)
            }
        }
    }

    return nameDeferred.await()
}
```

3. Opinion

- En general veo bastante bien este codelab, pero me he percatado de que la forma de guardar datos en local es mucho mas engorrosa que con el uso de otras tecnologias como *React Native* o *Flutter*.