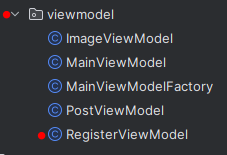
**ACTIVIDAD N°13 :**

**Clase Java de la vista: RegisterViewModel**

RegisterViewModel (registro):



**MÉTODOS**

1. **constructor**

private final MutableLiveData<String> registerResult = new MutableLiveData<>();

registerResult es la variable observable donde guardamos el resultado del registro (exitoso o fallido), se actualiza dependiendo de si el registro fue exitoso o si ocurrió un error.

public LiveData<String> getRegisterResult() {  
 return registerResult;  
}

Como es un “get” proporciona acceso al objeto registerResult como un LiveData, nos permite que otras partes del código (como una *Activity* o un *Fragment*) observen los cambios en el resultado del registro.

public void register(User user) {  
 // Crea un nuevo ParseUser  
 ParseUser parseUser = new ParseUser();  
 parseUser.setUsername(user.getUsername());  
 parseUser.setPassword(user.getPassword());  
 parseUser.setEmail(user.getEmail());  
  
 // Registra el usuario en Parse  
 parseUser.signUpInBackground(e -> {  
 if (e == null) {  
 Log.d("Registro", "Usuario registrado correctamente."+e);  
 registerResult.setValue("Registro exitoso");  
 } else {  
  
 registerResult.setValue("Error: " + e.getMessage());  
 }  
 });  
}

* Crea un objeto ParseUser utilizando la clase ParseUser del Parse SDK.
* Establece el nombre de usuario (setUsername), contraseña (setPassword) y correo electrónico (setEmail) utilizando los datos del objeto User.
* Usa el método signUpInBackground() de Parse para registrar al usuario en segundo plano.
* El registro ocurre de forma asíncrona, y el resultado es recibido mediante una función lambda (e -> { ... }):
  + Si e == null: Indica que el registro fue exitoso, se actualiza registerResult con el mensaje "Registro exitoso".
  + Si e != null: Indica un error en el proceso de registro, e.getMessage() contiene la descripción del error, que se almacena en registerResult.

**CÓDIGO**

import android.util.Log;  
import androidx.lifecycle.LiveData;  
import androidx.lifecycle.MutableLiveData;  
import androidx.lifecycle.ViewModel;  
import com.example.appchat.model.User;  
import com.parse.ParseUser;  
  
public class RegisterViewModel extends ViewModel {  
 private final MutableLiveData<String> registerResult = new MutableLiveData<>();  
  
 public LiveData<String> getRegisterResult() {  
 return registerResult;  
 }  
  
 public void register(User user) {  
 // Crea un nuevo ParseUser  
 ParseUser parseUser = new ParseUser();  
 parseUser.setUsername(user.getUsername());  
 parseUser.setPassword(user.getPassword());  
 parseUser.setEmail(user.getEmail());  
  
 // Registra el usuario en Parse  
 parseUser.signUpInBackground(e -> {  
 if (e == null) {  
 Log.d("Registro", "Usuario registrado correctamente."+e);  
 registerResult.setValue("Registro exitoso");  
 } else {  
  
 registerResult.setValue("Error: " + e.getMessage());  
 }  
 });  
 }  
}