Registro Diario de Avances – Sprint 1 Braille Express

**Fecha:** 01/05/2025

**Autor:** Ylia Jamile Ochoa Gutierrez

**Versión:** 1.0

# 1. Introducción

Este documento detalla el seguimiento diario del Sprint 1 del proyecto Braille Express, enfocado en el desarrollo de las funcionalidades clave para el módulo de registro y autenticación de usuarios. Se incluyen actividades diarias, tareas completadas, evidencias de código, un burndown chart y un análisis del rendimiento del equipo.

# 2. Lineamientos de Diseño

Desarrollar una primera versión funcional de la aplicación TaskManager que permita a los usuarios:

* Diseño de interfaces de registro y login.
* Registro seguro de usuarios (con almacenamiento en Firebase Realtime Database).
* Validación de DNI.

# 3. Historias de Usuario y Tareas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Backlog ID | Historia de Usuario | Tiempo Estimado | Fecha de Inicio | Fecha de Finalización |
| HU-1 | Registro de usuario | 5 pts | 21/04/2025 | 29/04/2025 |
| HU-2 | Inicio y cierre de sesión en la nube | 8 pts | 29/04/2025 | 30/04/2025 |

# 4. Evidencias de Código

## 4.1 Historia de Usuario 1 — Registro de Usuario

Archivo: UsuarioModel.java

public class UsuarioModel {

// Campos del modelo

private String nombre;

private String apellido;

private int edad;

private String genero;

private String dni;

private String password;

public UsuarioModel() {

}

//Firebase Realtime Database necesita getters y setters para mapear automáticamente los objetos Java ↔ JSON

public UsuarioModel(String nombre, String apellido, int edad, String genero, String dni, String password) {

this.nombre = nombre;

this.apellido = apellido;

this.edad = edad;

this.genero = genero;

this.dni = dni;

this.password = password;

}

public String getNombre() {

return nombre;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public String getApellido() {

return apellido;

}

public void setApellido(String apellido) {

this.apellido = apellido;

}

public int getEdad() {

return edad;

}

public void setEdad(int edad) {

this.edad = edad;

}

public String getGenero() {

return genero;

}

public void setGenero(String genero) {

this.genero = genero;

}

public String getDni() {

return dni;

}

public void setDni(String dni) {

this.dni = dni;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

}

Archivo: UsuarioRepositorio.java

public class UsuarioRepositorio {

// Referencia a la colección "usuarios" en Firebase

private final DatabaseReference usuariosRef;

public UsuarioRepositorio() {

FirebaseDatabase database = FirebaseDatabase.getInstance();

usuariosRef = database.getReference("usuarios");

}

//Registra un nuevo usuario en Firebase.

public void registrarUsuario(UsuarioModel usuario, String dni,

String password, OnUsuarioRegistradoListener listener) {

usuario.setPassword(password);

usuariosRef.child(dni).setValue(usuario)

.addOnSuccessListener(aVoid -> listener.onRegistrado())

.addOnFailureListener(listener::onError);

}

//Valida las credenciales de login comparando con la base de datos.

public void validarLogin(String dni, String password, OnLoginValidadoListener listener) {

usuariosRef.child(dni).get().addOnCompleteListener(task -> {

if (task.isSuccessful() && task.getResult().exists()) {

UsuarioModel usuario = task.getResult().getValue(UsuarioModel.class);

if (usuario != null && usuario.getPassword().equals(password)) {

listener.onLoginExitoso(usuario);

} else {

listener.onLoginFallido("Credenciales incorrectas");

}

} else {

listener.onLoginFallido("Usuario no encontrado");

}

}).addOnFailureListener(listener::onError);

}

//Listener para eventos de registro.

public interface OnUsuarioRegistradoListener {

void onRegistrado();

void onError(@NonNull Exception e);

}

//Listener para eventos de login.

public interface OnLoginValidadoListener {

void onLoginExitoso(UsuarioModel usuario);

void onLoginFallido(String mensajeError);

void onError(@NonNull Exception e);

}

}

Archivo: RegistroFragment.java

public class RegistroFragment extends Fragment {

private RegistroViewModel registroViewModel;

private FragmentRegistroBinding binding;

private TextInputEditText inputNombre, inputApellido, inputEdad, inputDNI, inputPassword;

private RadioGroup radioGroupGenero;

private Button btnRegistrar;

@Override

public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

// Configuración del View Binding

binding = FragmentRegistroBinding.inflate(inflater, container, false);

return binding.getRoot();

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

super.onViewCreated(view, savedInstanceState);

// Inicialización del ViewModel

registroViewModel = new ViewModelProvider(this).get(RegistroViewModel.class);

// Vinculación de vistas (alternativa con binding)

inputNombre = binding.Nombre;

inputApellido = binding.Apellidos;

inputEdad = binding.Edad;

inputDNI = binding.Dni;

inputPassword = binding.password;

radioGroupGenero = binding.radioGroupGenero;

btnRegistrar = binding.btnRegistrar;

// Evento de clic para registro

btnRegistrar.setOnClickListener(v -> registrarUsuario());

// Observadores del ViewModel

configurarObservadores();

}

//Recopila datos del formulario y solicita registro mediante

private void registrarUsuario() {

// Extracción de datos con validación básica

String nombre = inputNombre.getText().toString().trim();

String apellido = inputApellido.getText().toString().trim();

String edad = inputEdad.getText().toString().trim();

String dni = inputDNI.getText().toString().trim();

String password = inputPassword.getText().toString().trim();

// Validación de género seleccionado

int selectedGeneroId = radioGroupGenero.getCheckedRadioButtonId();

if (selectedGeneroId == -1) {

Toast.makeText(getContext(), "Seleccione un género", Toast.LENGTH\_SHORT).show();

return;

}

String genero = ((RadioButton) getView().findViewById(selectedGeneroId)).getText().toString();

// Envío de datos al ViewModel

registroViewModel.registrarUsuario(nombre, apellido, edad, genero, dni, password);

}

//Configura observadores para eventos del ViewModel:

private void configurarObservadores() {

registroViewModel.getRegistroExitoso().observe(getViewLifecycleOwn r(), exito -> {

if (exito != null && exito) {

Toast.makeText(getContext(), "Registro exitoso", Toast.LENGTH\_SHORT).show();

Navigation.findNavController(requireView()).navigate(R.id.action\_RegistroFragment\_to\_LoginFragment);

}

});

registroViewModel.getMensajeError().observe(getViewLifecycleOwner(), error -> {

if (!TextUtils.isEmpty(error)) {

Toast.makeText(getContext(), error, Toast.LENGTH\_SHORT).show();

}

});

}

@Override

public void onDestroyView() {

super.onDestroyView();

binding = null; // Limpieza del View Binding

}

}

Archivo: RegistroViewModel.java

public class RegistroViewModel extends ViewModel {

// LiveData para comunicación con la UI

private final MutableLiveData<Boolean> registroExitoso = new MutableLiveData<>();

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

// Provee el estado del registro a la UI

public LiveData<Boolean> getRegistroExitoso() {

return registroExitoso;

}

//Manda mensajes de error a la UI

public LiveData<String> getMensajeError() {

return mensajeError;

}

//Registra un nuevo usuario con validaciones y seguridad básica

public void registrarUsuario(String nombre, String apellido, String edad, String genero, String dni, String password) {

// Validación de campos obligatorios

if (TextUtils.isEmpty(nombre) || TextUtils.isEmpty(apellido) ||

TextUtils.isEmpty(edad) || TextUtils.isEmpty(genero) || TextUtils.isEmpty(dni)) {

mensajeError.setValue("Todos los campos son obligatorios.");

return;

}

// Validación adicional recomendada para producción:

if (password.length() < 6) {

mensajeError.setValue("La contraseña debe tener al menos 6 caracteres.");

return;

}

// Hash de la contraseña con BCrypt

String hashedPassword = BCrypt.withDefaults().hashToString(12, password.toCharArray());

// Configuración de Firebase

DatabaseReference database = FirebaseDatabase.getInstance().getReference("usuarios");

// Estructura de datos del usuario

HashMap<String, Object> datosUsuario = new HashMap<>();

datosUsuario.put("Nombre", nombre);

datosUsuario.put("Apellidos", apellido);

datosUsuario.put("Edad", edad);

datosUsuario.put("Genero", genero);

datosUsuario.put("Dni", dni);

datosUsuario.put("Password", hashedPassword); // Almacena solo el hash

// Operación de escritura en Firebase

database.child(dni).setValue(datosUsuario)

.addOnSuccessListener(unused -> registroExitoso.setValue(true))

.addOnFailureListener(e -> mensajeError.setValue("Error al registrar: " + e.getMessage()));

}

}

Archivo: fragment\_registro.xml

//Nombre

<TextView

android:id="@+id/titleRegister"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="4dp"

android:text="REGISTRO"

android:textColor="@color/black"

android:textSize="24sp"

android:textStyle="bold"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<TextView

android:id="@+id/labelNombre"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="28dp"

android:text="Nombre:"

android:textSize="16sp"

app:layout\_constraintTop\_toBottomOf="@+id/titleRegister"

tools:layout\_editor\_absoluteX="24dp" />

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/nombreLayout"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:layout\_constraintTop\_toBottomOf="@id/labelNombre"

android:layout\_marginTop="4dp">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/Nombre"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Ingrese su nombre" />

</com.google.android.material.textfield.TextInputLayout>

//Apellidos

<TextView

android:id="@+id/labelApellidos"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Apellidos:"

android:textSize="16sp"

android:layout\_marginTop="16dp"

app:layout\_constraintTop\_toBottomOf="@id/nombreLayout" />

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/apellidosLayout"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:layout\_constraintTop\_toBottomOf="@id/labelApellidos"

android:layout\_marginTop="4dp">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/Apellidos"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Ingrese sus apellidos" />

</com.google.android.material.textfield.TextInputLayout>

//Edad y Género

<LinearLayout

android:id="@+id/edadGeneroLayout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:layout\_marginTop="16dp"

app:layout\_constraintTop\_toBottomOf="@id/apellidosLayout">

<!-- Edad -->

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="start|center\_vertical"

android:layout\_marginEnd="8dp"

android:text="Edad:"

android:textSize="16sp" />

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/Edad"

android:layout\_width="95dp"

android:layout\_height="wrap\_content"

android:hint="Edad"

android:inputType="number" />

</LinearLayout>

<!-- Género -->

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:layout\_marginTop="8dp">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Género:"

android:textSize="16sp"

android:layout\_gravity="start|center\_vertical"

android:layout\_marginEnd="8dp"/>

<RadioGroup

android:id="@+id/radioGroupGenero"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<RadioButton

android:id="@+id/radioMasculino"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Masculino" />

<RadioButton

android:id="@+id/radioFemenino"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Femenino" />

</RadioGroup>

</LinearLayout>

</LinearLayout>

//DNI

<TextView

android:id="@+id/labelDniReg"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp"

android:text="DNI:"

android:textSize="16sp"

app:layout\_constraintTop\_toBottomOf="@id/edadGeneroLayout"

tools:layout\_editor\_absoluteX="24dp" />

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/dniRegLayout"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:layout\_constraintTop\_toBottomOf="@id/labelDniReg"

android:layout\_marginTop="4dp">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/Dni"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:inputType="number"

android:hint="Ingrese su DNI"

android:maxLength="8"/>

</com.google.android.material.textfield.TextInputLayout>

//Contraseña

<TextView

android:id="@+id/labelPasswordReg"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Contraseña:"

android:textSize="16sp"

android:layout\_marginTop="16dp"

app:layout\_constraintTop\_toBottomOf="@id/dniRegLayout" />

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/passwordRegLayout"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:layout\_constraintTop\_toBottomOf="@id/labelPasswordReg"

android:layout\_marginTop="4dp">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Cree una contraseña"

android:inputType="textPassword"/>

</com.google.android.material.textfield.TextInputLayout>

//Botón Registro

<Button

android:id="@+id/btnRegistrar"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="REGISTRARSE"

android:textColor="@color/white"

android:backgroundTint="@color/purple\_500"

android:layout\_marginTop="24dp"

app:layout\_constraintTop\_toBottomOf="@id/passwordRegLayout" />

//Enlace a login

<TextView

android:id="@+id/linkLogin"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp"

android:text="¿Ya tienes cuenta? Inicia sesión"

android:textColor="@color/purple\_500"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/btnRegistrar" />

</androidx.constraintlayout.widget.ConstraintLayout>

## 4.2 Historia de Usuario 2 — Inicio y cierre de sesión en la nube

Archivo: fragment\_login.xml

//Título "INICIO DE SESIÓN"

<TextView

android:id="@+id/titleLogin"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="16dp"

android:text="INICIO DE SESIÓN"

android:textColor="@color/black"

android:textSize="24sp"

android:textStyle="bold"

app:layout\_constraintBottom\_toTopOf="@+id/logoApp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

//Logo de la aplicación

<ImageView

android:id="@+id/logoApp"

android:layout\_width="150dp"

android:layout\_height="165dp"

android:layout\_marginTop="88dp"

android:src="@drawable/logo"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.497"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/titleLogin"

android:padding="2dp" />

//TextView DNI y campo para DNI

<TextView

android:id="@+id/labelDni"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="32dp"

android:text="DNI:"

android:textSize="16sp"

app:layout\_constraintTop\_toBottomOf="@id/logoApp"

tools:layout\_editor\_absoluteX="24dp" />

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/dniLayout"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:layout\_constraintTop\_toBottomOf="@id/labelDni"

android:layout\_marginTop="8dp">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/Dni"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Ingrese su DNI"

android:inputType="number"

android:maxLength="8"/> <!-- Validación de 8 dígitos -->

</com.google.android.material.textfield.TextInputLayout>

//TextView Contraseña y campo para Contraseña

<TextView

android:id="@+id/labelPassword"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="16dp"

android:text="Contraseña:"

android:textSize="16sp"

app:layout\_constraintTop\_toBottomOf="@id/dniLayout"

tools:layout\_editor\_absoluteX="24dp" />

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/passwordLayout"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:layout\_constraintTop\_toBottomOf="@id/labelPassword"

android:layout\_marginTop="8dp">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Ingrese su contraseña"

android:inputType="textPassword"/>

</com.google.android.material.textfield.TextInputLayout>

//Botón de Inicio de Sesión

<Button

android:id="@+id/btnLogin"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="24dp"

android:backgroundTint="@color/purple\_500"

android:text="INICIAR SESIÓN"

app:layout\_constraintTop\_toBottomOf="@+id/passwordLayout"

tools:layout\_editor\_absoluteX="24dp"

android:textColor="@color/white" />

//Link de Registro

<TextView

android:id="@+id/linkRegistro"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="¿No tienes cuenta? Regístrate aquí"

android:textColor="@color/purple\_500"

app:layout\_constraintTop\_toBottomOf="@id/btnLogin"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

android:layout\_marginTop="24dp"/>

</androidx.constraintlayout.widget.ConstraintLayout>

# 5. Burndown Chart

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Backlog ID | Historia de Usuario | Tiempo Estimado | 21/04/2025 | 22/04/2025 | 23/04/2025 | 24/04/2025 | 25/04/2025 | 26/04/2025 |
| **Dia1** | **Dia2** | **Dia3** | **Dia4** | **Dia5** | **Dia6** |
| HU-1 | Registro de usuario | 54 | 10 | 6 | 11 | 8 | 9 | 10 |
| HU-2 | Inicio y cierre de sesión en la nube | 7 |  |  |  | 4 | 2 | 1 |
| Tiempo de trabajo | | 61 | 10 | 6 | 11 | 12 | 11 | 11 |

# 6. Análisis del Sprint

* Se cumplió con todas las tareas estimadas.
* El progreso fue constante y sin bloqueos mayores.
* Las decisiones técnicas sobre persistencia con localStorage resultaron efectivas para este primer Sprint.
* Se logró modularizar el código y documentarlo en comentarios.
* Las estimaciones casi precisas y se logró un buen ritmo de trabajo.

# 7. Conclusiones

El Sprint 1 de *Braille Express* culminó exitosamente con la implementación completa de las funcionalidades básicas de registro y autenticación, cumpliendo los objetivos planteados dentro del cronograma estimado. El equipo demostró eficiencia en la adopción de Firebase Realtime Database y BCrypt para garantizar seguridad básica, así como en la creación de interfaces accesibles siguiendo pautas WCAG.

La arquitectura MVVM modularizada y la documentación detallada del código sentaron las bases para futuras iteraciones, como la integración con Firebase Authentication en el próximo sprint. La gestión ágil permitió resolver riesgos menores sin afectar la productividad, evidenciando un equilibrio óptimo entre calidad y velocidad de entrega.