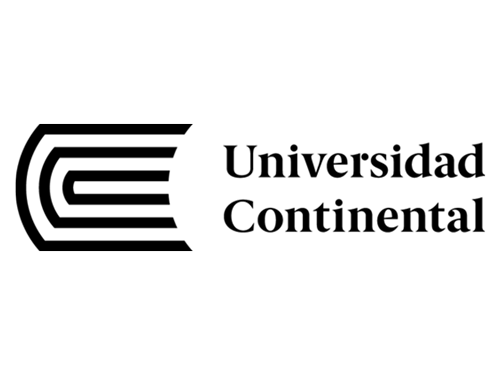
***"Año de la recuperación y consolidación de la economía peruana"***



**FACULTAD DE INGENIERÍA**

**TEMA:**

**TP2.E8 Entrega de sprint 01**

**Curso:**

**Taller de proyectos 2**

**NRC: 17185**

**Docente:**

**Americo Estrada Sanchez**

**Alumno:**

* [**CHRISTIAN CASTRO USTO**](mailto:73024009@continental.edu.pe)

**CUSCO -  PERÚ**

**2025**

Sprint 01 - SLEEPAPP

**Fecha:** 29/04/2025

**Autor: Christian Castro Usto**

**Versión:** 1.0

Sprint 01 – SleepApp

# 1. Introducción

Este documento detalla el desarrollo del Sprint 01 del proyecto SleepApp, una aplicación móvil orientada a mejorar la calidad del sueño de los usuarios mediante el registro de rutinas y alarmas personalizadas. En este sprint se trabajó en las funcionalidades fundamentales de acceso: registro de usuario e inicio de sesión, incluyendo interfaces, validaciones y soporte offline. También se documentan los avances técnicos mediante evidencias de código y el seguimiento de tareas a través del burndown chart.

# 2. Objetivo del Sprint

Implementar funcionalidades básicas de acceso del usuario: registro e inicio de sesión, incluyendo persistencia de datos local y validaciones.

# 3. Historias de Usuario y Tareas

* Historia de Usuario 1.1: Registro de usuario

Descripción: Como usuario, quiero poder registrarme en la aplicación proporcionando mi correo y contraseña, para poder acceder a mis datos y configuraciones de sueño en cualquier dispositivo.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Tarea | Responsable | Horas |
| 1.1.1 | Diseñar la interfaz de formulario de ingreso de datos de usuario | UX/UI Designer | 4 |
| 1.1.2 | Implementar el formulario de ingreso de datos de usuario en la app | Frontend Dev | 6 |
| 1.1.3 | Configurar la conexión con la base de datos para el registro | Backend Dev | 6 |
| 1.1.4 | Implementar validaciones de formulario | Backend Dev | 5 |
| 1.1.5 | Implementar funcionalidad offline para el registro (guardado en local) | Backend Dev | 6 |
| 1.1.6 | Pruebas unitarias y de integración | QA Engineer | 6 |

* Historia de Usuario 1.2: Iniciar sesión

Descripción: Como usuario, quiero poder iniciar sesión con mi correo y contraseña, para acceder a mis datos en la aplicación.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Tarea | Responsable | Horas |
| 1.2.1 | Diseñar la interfaz de inicio de sesión | UX/UI Designer | 4 |
| 1.2.2 | Implementar el formulario de inicio de sesión en la app | Frontend Dev | 6 |
| 1.2.3 | Configurar Sqlite para consultar los datos del usuario login | Backend Dev | 6 |
| 1.2.4 | Implementar validaciones (correo válido, contraseña correcta) | Frontend Engineer | 5 |
| 1.2.5 | Implementar inicio de sesión sin internet | Backend Dev | 6 |

# 4. Evidencias de Código

A continuación, se incluirán fragmentos del código implementado para las historias de usuario.

# 4.1 Historia de Usuario 01

* Diseñar la interfaz de formulario de ingreso de datos de usuario

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@drawable/fondo"  
 tools:context=".MainActivity">  
  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center"  
 android:layout\_margin="10dp"  
 android:layout\_weight="1"  
 android:gravity="center"  
 android:text="CREAR NUEVA CUENTA"  
 android:textColor="#000000"  
 android:textSize="20sp"  
 android:textStyle="bold" />  
 <EditText  
 android:id="@+id/etDni"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="10dp"  
 android:layout\_weight="1"  
 android:hint="DNI"  
 android:inputType="textEmailAddress"/>  
 <EditText  
 android:id="@+id/etUsuario"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="10dp"  
 android:layout\_weight="1"  
 android:hint="Usuario"  
 android:inputType="textEmailAddress"/>  
 <EditText  
 android:id="@+id/etContraseña"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="10dp"  
 android:layout\_weight="1"  
 android:hint="Contraseña"  
 android:inputType="textPassword"/>  
 <EditText  
 android:id="@+id/etNombre"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="10dp"  
 android:layout\_weight="1"  
 android:hint="Nombre"  
 android:inputType="textPersonName"/>  
 <EditText  
 android:id="@+id/etApellido"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="10dp"  
 android:layout\_weight="1"  
 android:hint="Apellidos"  
 android:inputType="textPersonName"/>  
 <EditText  
 android:id="@+id/etCorreo"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="10dp"  
 android:layout\_weight="1"  
 android:hint="Correo"  
 android:inputType="textEmailAddress"/>  
  
 <EditText  
 android:id="@+id/etFechanacimiento"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="10dp"  
 android:layout\_weight="1"  
 android:hint="FechaNacimiento"  
 android:inputType=" "/>  
  
  
 <Button  
 android:id="@+id/btnRegRegistrar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="7dp"  
 android:layout\_weight="1"  
 android:backgroundTint="#12FF00"  
 android:text="Registrarse"  
 android:textColor="#FFFFFF"  
 android:textSize="30dp"  
 android:textStyle="bold" />  
 <Button  
 android:id="@+id/btnRegCancelar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_margin="7dp"  
 android:layout\_weight="1"  
 android:backgroundTint="#F44336"  
 android:text="Cancelar"  
 android:textColor="#ffffff"  
 android:textSize="30dp"  
 android:textStyle="bold"/>  
  
</LinearLayout>

* Implementar el formulario de ingreso de datos de usuario en la app

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

* Configurar la conexión con la base de datos para el registro
* // Insertar en la base de datos si las validaciones son correctas  
  try {  
   SQLiteOpenHelper conexion = new SqlLiteHelper(getApplicationContext());  
   SQLiteDatabase baseDeDatos = conexion.getWritableDatabase();  
   ContentValues valores = new ContentValues();  
   valores.put("dni", Integer.*valueOf*(dniStr));  
   valores.put("usuario", usuarioStr);  
   valores.put("contraseña", contraseñaStr);  
   valores.put("nombre", nombreStr);  
   valores.put("apellido", apellidoStr);  
   valores.put("correo", correoStr);  
   valores.put("fechanacimiento", fechanacimientoStr);  
    
   long resultado = baseDeDatos.insert("tusuarios", null, valores);  
   if (resultado != -1) {  
   Toast.*makeText*(getApplicationContext(), "Se guardó correctamente", Toast.*LENGTH\_SHORT*).show();  
   } else {  
   Toast.*makeText*(getApplicationContext(), "Error al guardar el registro", Toast.*LENGTH\_SHORT*).show();  
   }  
    
  } catch (SQLException e) {  
   Toast.*makeText*(getApplicationContext(), "Error: " + e.getMessage(), Toast.*LENGTH\_SHORT*).show();  
  }
* Implementar validaciones de formulario

// Validación de los campos  
if (TextUtils.*isEmpty*(dniStr) || dniStr.length() != 8 || !TextUtils.*isDigitsOnly*(dniStr)) {  
 Toast.*makeText*(getApplicationContext(), "DNI debe ser de 8 dígitos numéricos", Toast.*LENGTH\_SHORT*).show();  
 return;  
}  
  
if (TextUtils.*isEmpty*(contraseñaStr) || contraseñaStr.length() < 6) {  
 Toast.*makeText*(getApplicationContext(), "La contraseña debe tener al menos 6 caracteres", Toast.*LENGTH\_SHORT*).show();  
 return;  
}  
  
if (TextUtils.*isEmpty*(usuarioStr) || TextUtils.*isEmpty*(nombreStr) || TextUtils.*isEmpty*(apellidoStr)  
 || TextUtils.*isEmpty*(correoStr) || TextUtils.*isEmpty*(fechanacimientoStr)) {  
 Toast.*makeText*(getApplicationContext(), "Todos los campos deben estar llenos", Toast.*LENGTH\_SHORT*).show();  
 return;  
}

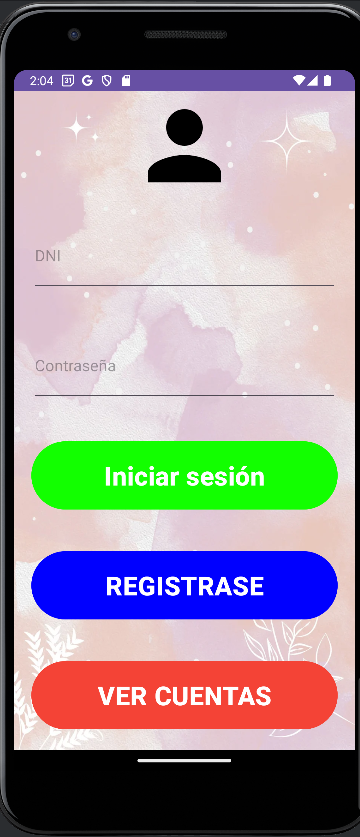
* Implementar funcionalidad offline para el registro (guardado en local)
* @Override  
  public void onCreate(SQLiteDatabase db) {  
   // Crear tabla tusuarios  
   db.execSQL("CREATE TABLE IF NOT EXISTS " + *TABLA\_USUARIOS* + " (" +  
   *COLUMNA\_DNI* + " INTEGER PRIMARY KEY, " +  
   *COLUMNA\_USUARIO* + " TEXT NOT NULL, " +  
   *COLUMNA\_CONTRASENA* + " TEXT NOT NULL, " +  
   *COLUMNA\_NOMBRE* + " TEXT NOT NULL, " +  
   *COLUMNA\_APELLIDO* + " TEXT NOT NULL, " +  
   *COLUMNA\_CORREO* + " TEXT NOT NULL, " +  
   *COLUMNA\_FECHA\_NACIMIENTO* + " TEXT NOT NULL)");
* Pruebas unitarias y de integración

Texto

Descripción generada automáticamente

# 4.1 Historia de Usuario 02

* Diseñar la interfaz de inicio de sesión
* <?xml version="1.0" encoding="utf-8"?>  
  <LinearLayout  
   xmlns:android="http://schemas.android.com/apk/res/android"  
   xmlns:app="http://schemas.android.com/apk/res-auto"  
   xmlns:tools="http://schemas.android.com/tools"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   android:orientation="vertical"  
   tools:context=".MainActivity"  
   android:background="@drawable/fondo">  
    
   <ImageView  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   android:layout\_weight="1"  
   android:src="@drawable/ic\_user" />  
    
   <EditText  
   android:id="@+id/User"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   android:layout\_margin="20dp"  
   android:layout\_weight="1"  
   android:hint="DNI"  
   android:inputType="textEmailAddress" />  
    
   <EditText  
   android:id="@+id/Pass"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   android:layout\_margin="20dp"  
   android:layout\_weight="1"  
   android:hint="Contraseña"  
   android:inputType="textPassword"/>  
    
   <Button  
   android:id="@+id/btnInicio"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   android:layout\_margin="20dp"  
   android:layout\_weight="1"  
   android:backgroundTint="#12FF00"  
   android:text="Iniciar sesión"  
   android:textColor="#FFFFFF"  
   android:textSize="30dp"  
   android:textStyle="bold" />  
   <Button  
   android:id="@+id/btnRegistro"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   android:layout\_margin="20dp"  
   android:layout\_weight="1"  
   android:backgroundTint="#0000ff"  
   android:text="REGISTRASE"  
   android:textColor="#ffffff"  
   android:textSize="30dp"  
   android:textStyle="bold"/>  
   <Button  
   android:id="@+id/btnVerCuentas"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   android:layout\_margin="20dp"  
   android:layout\_weight="1"  
   android:backgroundTint="#F44336"  
   android:text="VER CUENTAS"  
   android:textColor="#ffffff"  
   android:textSize="30dp"  
   android:textStyle="bold"/>  
    
  </LinearLayout>
* Implementar el formulario de inicio de sesión en la app



* Configurar Sqlite para consultar los datos del usuario login
* String dniIngresado = dni.getText().toString();  
  String contraseñaIngresada = contraseña.getText().toString();
* Implementar validaciones (correo válido, contraseña correcta)

if (cursor.moveToFirst()) {  
 // Inicio de sesión exitoso  
 Toast.*makeText*(getApplicationContext(), "Inicio de sesión exitoso", Toast.*LENGTH\_SHORT*).show();  
  
 // Guardar el DNI del usuario actual para futuras consultas  
 String dniUsuarioActual = cursor.getString(cursor.getColumnIndex("dni"));  
 Intent ventanaPrincipal = new Intent(MainActivity.this, MenuPrincipal.class);  
 ventanaPrincipal.putExtra("dniUsuarioActual", dniUsuarioActual); // Pasar el DNI a la siguiente actividad  
 startActivity(ventanaPrincipal);  
 } else {  
 // Usuario o contraseña incorrectos  
 Toast.*makeText*(getApplicationContext(), "DNI o contraseña incorrectos", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 cursor.close();  
 baseDeDatos.close();  
 }  
});

* Implementar inicio de sesión sin internet
* @Override  
  public void onClick(View v) {  
   SQLiteOpenHelper conexion = new SqlLiteHelper(getApplicationContext());  
   SQLiteDatabase baseDeDatos = conexion.getReadableDatabase();

# 5. Burndown Chart

El siguiente gráfico representa el progreso del sprint, comparando las horas restantes con los días de ejecución.

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente con confianza media

# 6. Análisis del Sprint

# Durante el Sprint 01, el equipo mostró un buen desempeño en el diseño de interfaces de registro e inicio de sesión, destacando la coordinación entre diseño y desarrollo frontend. Sin embargo, se presentaron dificultades en la implementación de funcionalidades offline, especialmente en la gestión de datos locales con SQLite. También se evidenció la necesidad de mejorar la planificación en cuanto a estimaciones de tiempo y definición precisa de los criterios de aceptación. Para el siguiente sprint, se priorizará una revisión más detallada de dependencias técnicas y casos de prueba más específicos para evitar bloqueos en el desarrollo.

# 7. Conclusiones

El Sprint 01 permitió establecer una base sólida para las funcionalidades de acceso del usuario, cumpliendo con el diseño y parte de la implementación del registro e inicio de sesión. El equipo trabajó de forma colaborativa y organizada, sentando las bases para abordar con mayor eficacia las funcionalidades de monitoreo del sueño en los próximos sprints.