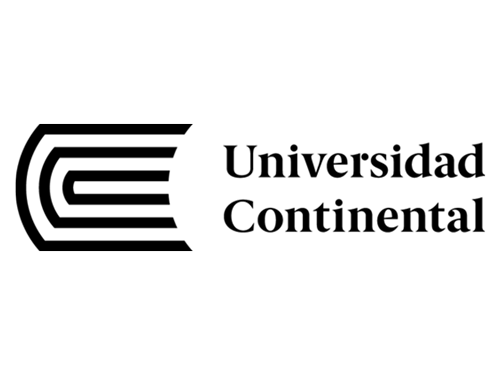
***"Año de la recuperación y consolidación de la economía peruana"***



**FACULTAD DE INGENIERÍA**

**TEMA:**

**TP2.E8 Entrega de sprint 02**

**Curso:**

**Taller de proyectos 2**

**NRC: 17185**

**Docente:**

**Americo Estrada Sanchez**

**Alumno:**

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

**CUSCO -  PERÚ**

**2025**

Sprint 02 - SLEEPAPP

**Fecha:** 14/04/2025

**Autor: Christian Castro Usto**

**Versión:** 1.0

Sprint 02 – SleepApp

# 1. Introducción

Durante el **Sprint 02** del proyecto **SleepApp**, se continuará con el desarrollo de funcionalidades clave que permitirán avanzar hacia una versión más completa de la aplicación. En este sprint se dará continuidad a la Historia de Usuario 1.2, añadiendo pruebas de integración para asegurar la correcta funcionalidad del inicio de sesión. Además, se incorporarán nuevas historias de usuario enfocadas en la **gestión de la rutina diaria del usuario**, permitiendo registrar, modificar y validar actividades como estudio, trabajo, recreación y descanso. Esta información será esencial para que la aplicación pueda analizar los hábitos del usuario y posteriormente generar recomendaciones personalizadas sobre el sueño. También se iniciará el desarrollo de la funcionalidad de **configuración de alarmas personalizadas**, que permitirá a los usuarios definir alarmas de acuerdo con su rutina planificada.

# 3. Lineamientos de Diseño

Durante el **Sprint 02**, se desarrollará una versión más completa de la aplicación móvil **SleepApp**, enfocada en incorporar funcionalidades relacionadas con la **rutina diaria del usuario** y la **configuración de alarmas personalizadas**. El objetivo de este sprint será permitir que los usuarios gestionen su planificación diaria (horas de estudio, trabajo, recreación y descanso), y que la aplicación pueda utilizar esta información para futuras recomendaciones sobre el sueño:

* Registro y gestión de rutina diaria del usuario
* Edición y modificación de la rutina registrada
* Almacenamiento local de datos para permitir el uso sin conexión
* Diseño de interfaces para rutina y alarma

Estas funcionalidades permitirán que en los siguientes sprints se integre el módulo de análisis de patrones de sueño, alarmas inteligentes, y recomendaciones personalizadas, basadas en los datos que el usuario haya proporcionado a través de estas rutinas.

# 2. Objetivo del Sprint

Se implementarán las funcionalidades básicas de acceso del usuario, que incluyen el registro y el inicio de sesión, junto con la persistencia de datos de forma local mediante SQLite y la incorporación de validaciones necesarias para garantizar la integridad de los datos ingresados.

# 3. Historias de Usuario y Tareas

* Historia de Usuario 2.1: Registro y gestión de rutinas

Descripción: Como usuario, quiero registrar y gestionar mi rutina, incluyendo horas de estudio, recreación, trabajo y descanso, para que la aplicación pueda analizar mis patrones de sueño y recomendar mejoras.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Tarea | Responsable | Horas |
| 2.1.1 | Diseñar la interfaz de ingreso y edición de rutina | UX/UI Designer | 5 |
| 2.1.2 | Implementar formulario para ingresar rutina diaria | Frontend Dev | 6 |
| 2.1.3 | Crear validación en los tiempos seleccionados | Backend Dev | 4 |
| 2.1.4 | Implementar almacenamiento local para funcionamiento sin conexión. | Backend Dev | 4 |
| 2.1.5 | Pruebas unitarias y de integración | Backend Dev | 6 |

* Historia de Usuario 2.2: Modificar una rutina

Descripción: **Como usuario, quiero modificar mi rutina diaria para actualizar cambios en mi planificación**.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Tarea | Responsable | Horas |
| 2.2.1 | Diseñar la interfaz de edición de rutina | UX/UI Designer | 4 |
| 2.2.2 | Agregar validaciones para la modificación de la rutina | Frontend Dev | 4 |
| 2.2.3 | Agregar opción para cancelar cambios antes de guardar | Backend Dev | 4 |
| 2.2.4 | Implementar conexión al almacenamiento local para funcionamiento sin conexión. | Frontend Engineer | 6 |
| 2.2.5 | Pruebas unitarias y de integración | 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

* Pruebas de integración

Captura de pantalla de computadora

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

# 4.1 Historia de Usuario 02

* Diseñar la interfaz de ingreso y edición de rutina usuario
* <?xml version="1.0" encoding="utf-8"?>  
  <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   android:padding="16dp"  
   android:background="@drawable/fondo"  
   >  
    
   <LinearLayout  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:orientation="vertical"  
   android:background="@drawable/fondo">  
    
   <TextView  
   android:id="@+id/textViewTitle"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="Registro de Actividades"  
   android:textSize="24sp"  
   android:textStyle="bold"  
   android:gravity="center"  
   android:layout\_gravity="center"/>  
    
   <!-- Edad con un Spinner para seleccionar un rango de edades -->  
   <TextView  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="Edad"/>  
    
   <Spinner  
   android:id="@+id/spinnerAge"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content" />  
    
   <!-- Horas de trabajo con SeekBar -->  
   <TextView  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="Horas de trabajo/estudio (h)" />  
    
   <SeekBar  
   android:id="@+id/seekBarWorkHours"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:max="16"/>  
    
   <TextView  
   android:id="@+id/textViewWorkHoursValue"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="0 h"/>  
    
   <!-- Tiempo de ejercicio con SeekBar -->  
   <TextView  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="Tiempo de ejercicio (min)" />  
    
   <SeekBar  
   android:id="@+id/seekBarExercise"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:max="120" />  
    
   <TextView  
   android:id="@+id/textViewExerciseValue"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="0 min"/>  
    
   <!-- Actividades recreativas con SeekBar (en horas) -->  
   <TextView  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:visibility="gone"  
   android:text="Tiempo de actividades recreativas (h)" />  
    
   <SeekBar  
   android:id="@+id/seekBarRecreationalActivities"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:visibility="gone"  
   android:max="10" />  
    
   <TextView  
   android:id="@+id/textViewRecreationalActivitiesValue"  
   android:layout\_width="wrap\_content"  
   android:visibility="gone"  
   android:layout\_height="wrap\_content"  
   android:text="0 h"/>  
    
   <!-- Consumo de café o estimulantes con SeekBar -->  
   <TextView  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="Consumo de café o estimulantes (tazas)" />  
    
   <SeekBar  
   android:id="@+id/seekBarCaffeine"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:max="10" />  
    
   <TextView  
   android:id="@+id/textViewCaffeineValue"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="0 tazas"/>  
    
   <!-- Uso de pantallas con SeekBar -->  
   <TextView  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="Uso de pantallas (h)" />  
    
   <SeekBar  
   android:id="@+id/seekBarScreenTime"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:max="12" />  
    
   <TextView  
   android:id="@+id/textViewScreenTimeValue"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="0 h"/>  
    
   <!-- Niveles de estrés o ansiedad con SeekBar -->  
   <TextView  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="Niveles de estrés o ansiedad (1-10)" />  
    
   <SeekBar  
   android:id="@+id/seekBarStress"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:max="10" />  
    
   <TextView  
   android:id="@+id/textViewStressValue"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="0"/>  
    
   <!-- Botón de guardar -->  
   <Button  
   android:id="@+id/btnSubmit"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:text="Guardar" />  
    
   <Button  
   android:id="@+id/btnCancelar"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:text="Cancelar"  
   android:backgroundTint="#F44336"/>  
    
    
   </LinearLayout>  
  </ScrollView>
* Implementar formulario para ingresar rutina diaria

Pantalla de celular con imagen de la pantalla de un computador

Descripción generada automáticamente con confianza media

* Configurar la conexión con la base de datos para el registro
* // Crear tabla trutina  
  db.execSQL("CREATE TABLE IF NOT EXISTS " + *TABLA\_RUTINA* + " (" +  
   *COLUMNA\_DNI* + " INTEGER, " +  
   *COLUMNA\_EDAD* + " TEXT NOT NULL, " +  
   *COLUMNA\_HORAS\_TRABAJO* + " INTEGER NOT NULL, " +  
   *COLUMNA\_EJERCICIO\_MINUTOS* + " INTEGER NOT NULL, " +  
   *COLUMNA\_RECREACION\_MINUTOS* + " INTEGER NOT NULL, " +  
   *COLUMNA\_CAFE\_CONSUMO* + " INTEGER NOT NULL, " + // Café  
   *COLUMNA\_TIEMPO\_PANTALLA* + " INTEGER NOT NULL, " + // Tiempo frente a pantallas  
   *COLUMNA\_ESTRES* + " INTEGER NOT NULL, " + // Estrés  
   "FOREIGN KEY (" + *COLUMNA\_DNI* + ") REFERENCES " + *TABLA\_USUARIOS* + "(" + *COLUMNA\_DNI* + ") ON DELETE CASCADE)");
* Crear validación en los tiempos seleccionados

// Conectar con los elementos del XML  
spinnerAge = findViewById(R.id.*spinnerAge*);  
seekBarWorkHours = findViewById(R.id.*seekBarWorkHours*);  
seekBarExercise = findViewById(R.id.*seekBarExercise*);  
seekBarRecreationalActivities = findViewById(R.id.*seekBarRecreationalActivities*);  
seekBarCaffeine = findViewById(R.id.*seekBarCaffeine*);  
seekBarScreenTime = findViewById(R.id.*seekBarScreenTime*);  
seekBarStress = findViewById(R.id.*seekBarStress*);  
textViewWorkHoursValue = findViewById(R.id.*textViewWorkHoursValue*);  
textViewExerciseValue = findViewById(R.id.*textViewExerciseValue*);  
textViewRecreationalActivitiesValue = findViewById(R.id.*textViewRecreationalActivitiesValue*);  
textViewCaffeineValue = findViewById(R.id.*textViewCaffeineValue*);  
textViewScreenTimeValue = findViewById(R.id.*textViewScreenTimeValue*);  
textViewStressValue = findViewById(R.id.*textViewStressValue*);  
btnSubmit = findViewById(R.id.*btnSubmit*);  
btnCancel = findViewById(R.id.*btnCancelar*);

* Implementar funcionalidad offline para el registro (guardado en local)
* // Manejar el botón de guardar  
  btnSubmit.setOnClickListener(new View.OnClickListener() {  
   @Override  
   public void onClick(View view) {  
   // Obtener los valores de las SeekBars  
   int workHours = seekBarWorkHours.getProgress();  
   int exerciseTime = seekBarExercise.getProgress();  
   int recreationalActivitiesTime = seekBarRecreationalActivities.getProgress();  
   int caffeineConsumption = seekBarCaffeine.getProgress();  
   int screenTime = seekBarScreenTime.getProgress();  
   int stressLevel = seekBarStress.getProgress();  
    
     
    
   // Guardar en la base de datos  
   sqlLiteHelper.insertarRutina(  
   dni,  
   spinnerAge.getSelectedItem().toString(),  
   workHours,  
   exerciseTime,  
   recreationalActivitiesTime,  
   caffeineConsumption,  
   screenTime,  
   stressLevel  
   );  
    
   Toast.*makeText*(registrorutina.this, "Datos guardados correctamente", Toast.*LENGTH\_SHORT*).show();  
   finish(); // Cerrar la actividad  
   }  
  });
* Pruebas unitarias y de integración

Captura de pantalla de computadora

Descripción generada automáticamente

# 4.1 Historia de Usuario 2.2: Modificar una rutina

* Diseñar la interfaz de edición de rutina
* <?xml version="1.0" encoding="utf-8"?>  
  <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   android:padding="16dp"  
   android:background="@drawable/fondo"  
   >  
    
   <LinearLayout  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:orientation="vertical"  
   android:background="@drawable/fondo">  
    
   <TextView  
   android:id="@+id/textViewTitle"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="Registro de Actividades"  
   android:textSize="24sp"  
   android:textStyle="bold"  
   android:gravity="center"  
   android:layout\_gravity="center"/>  
    
   <!-- Edad con un Spinner para seleccionar un rango de edades -->  
   <TextView  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="Edad"/>  
    
   <Spinner  
   android:id="@+id/spinnerAge"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content" />  
    
   <!-- Horas de trabajo con SeekBar -->  
   <TextView  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="Horas de trabajo/estudio (h)" />  
    
   <SeekBar  
   android:id="@+id/seekBarWorkHours"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:max="16"/>  
    
   <TextView  
   android:id="@+id/textViewWorkHoursValue"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="0 h"/>  
    
   <!-- Tiempo de ejercicio con SeekBar -->  
   <TextView  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="Tiempo de ejercicio (min)" />  
    
   <SeekBar  
   android:id="@+id/seekBarExercise"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:max="120" />  
    
   <TextView  
   android:id="@+id/textViewExerciseValue"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="0 min"/>  
    
   <!-- Actividades recreativas con SeekBar (en horas) -->  
   <TextView  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:visibility="gone"  
   android:text="Tiempo de actividades recreativas (h)" />  
    
   <SeekBar  
   android:id="@+id/seekBarRecreationalActivities"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:visibility="gone"  
   android:max="10" />  
    
   <TextView  
   android:id="@+id/textViewRecreationalActivitiesValue"  
   android:layout\_width="wrap\_content"  
   android:visibility="gone"  
   android:layout\_height="wrap\_content"  
   android:text="0 h"/>  
    
   <!-- Consumo de café o estimulantes con SeekBar -->  
   <TextView  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="Consumo de café o estimulantes (tazas)" />  
    
   <SeekBar  
   android:id="@+id/seekBarCaffeine"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:max="10" />  
    
   <TextView  
   android:id="@+id/textViewCaffeineValue"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="0 tazas"/>  
    
   <!-- Uso de pantallas con SeekBar -->  
   <TextView  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="Uso de pantallas (h)" />  
    
   <SeekBar  
   android:id="@+id/seekBarScreenTime"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:max="12" />  
    
   <TextView  
   android:id="@+id/textViewScreenTimeValue"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="0 h"/>  
    
   <!-- Niveles de estrés o ansiedad con SeekBar -->  
   <TextView  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="Niveles de estrés o ansiedad (1-10)" />  
    
   <SeekBar  
   android:id="@+id/seekBarStress"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:max="10" />  
    
   <TextView  
   android:id="@+id/textViewStressValue"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="0"/>  
    
   <!-- Botón de guardar -->  
   <Button  
   android:id="@+id/btnSubmit"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:text="Guardar" />  
    
   <Button  
   android:id="@+id/btnCancelar"  
   android:layout\_width="match\_parent"  
   android:layout\_height="wrap\_content"  
   android:text="Cancelar"  
   android:backgroundTint="#F44336"/>  
    
    
   </LinearLayout>  
  </ScrollView>
* Agregar validaciones para la modificación de la rutina
* // Conectar con los elementos del XML  
  spinnerAge = findViewById(R.id.*spinnerAge*);  
  seekBarWorkHours = findViewById(R.id.*seekBarWorkHours*);  
  seekBarExercise = findViewById(R.id.*seekBarExercise*);  
  seekBarRecreationalActivities = findViewById(R.id.*seekBarRecreationalActivities*);  
  seekBarCaffeine = findViewById(R.id.*seekBarCaffeine*);  
  seekBarScreenTime = findViewById(R.id.*seekBarScreenTime*);  
  seekBarStress = findViewById(R.id.*seekBarStress*);  
  textViewWorkHoursValue = findViewById(R.id.*textViewWorkHoursValue*);  
  textViewExerciseValue = findViewById(R.id.*textViewExerciseValue*);  
  textViewRecreationalActivitiesValue = findViewById(R.id.*textViewRecreationalActivitiesValue*);  
  textViewCaffeineValue = findViewById(R.id.*textViewCaffeineValue*);  
  textViewScreenTimeValue = findViewById(R.id.*textViewScreenTimeValue*);  
  textViewStressValue = findViewById(R.id.*textViewStressValue*);  
  btnSubmit = findViewById(R.id.*btnSubmit*);  
  btnCancel = findViewById(R.id.*btnCancelar*);
* Agregar opción para cancelar cambios antes de guardar
* btnCancel.setOnClickListener(new View.OnClickListener() {  
   @Override  
   public void onClick(View view) {  
   //Volver a la pantalla principal  
   Intent Volver = new Intent(registrorutina.this,MenuPrincipal.class);  
   startActivity(Volver);  
   }  
  });
* Implementar conexión al almacenamiento local para funcionamiento sin conexión.

// Manejar el botón de guardar  
btnSubmit.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 // Obtener los valores de las SeekBars  
 int workHours = seekBarWorkHours.getProgress();  
 int exerciseTime = seekBarExercise.getProgress();  
 int recreationalActivitiesTime = seekBarRecreationalActivities.getProgress();  
 int caffeineConsumption = seekBarCaffeine.getProgress();  
 int screenTime = seekBarScreenTime.getProgress();  
 int stressLevel = seekBarStress.getProgress();

cursor.close();  
 baseDeDatos.close();  
 }  
});

// Guardar en la base de datos  
 sqlLiteHelper.insertarRutina(  
 dni,  
 spinnerAge.getSelectedItem().toString(),  
 workHours,  
 exerciseTime,  
 recreationalActivitiesTime,  
 caffeineConsumption,  
 screenTime,  
 stressLevel  
 );  
  
 Toast.*makeText*(registrorutina.this, "Datos guardados correctamente", Toast.*LENGTH\_SHORT*).show();  
 finish(); // Cerrar la actividad  
 }  
});

* Pruebas unitarias y de integración

Captura de pantalla de computadora

Descripción generada automáticamente

# 4.1 Historia de Usuario 03

* Diseñar la interfaz de configuración de alarma**.**
* <LinearLayout  
   xmlns:android="http://schemas.android.com/apk/res/android"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   android:orientation="vertical"  
   android:padding="16dp"  
   android:background="@drawable/fondo">  
    
    
   <!-- Selector de la hora para la alarma -->  
   <TimePicker  
   android:id="@+id/timePickerAlarm"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:timePickerMode="clock"  
   android:layout\_gravity="center"/>  
    
   <!-- Texto que muestra la hora sugerida para acostarse -->  
   <TextView  
   android:id="@+id/tvRecommendedSleepTime"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="Hora sugerida para acostarse"  
   android:layout\_gravity="center"  
   android:padding="10dp"/>  
    
   <!-- Texto que muestra la hora para dormir máxima -->  
   <TextView  
   android:id="@+id/tvMaxSleepTime"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="Hora para dormir máxima"  
   android:layout\_gravity="center"  
   android:padding="10dp"/>  
    
   <!-- Botón para seleccionar el tono de alarma -->  
   <Button  
   android:id="@+id/btnSelectTone"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="Seleccionar Tono de Alarma"  
   android:layout\_gravity="center"/>  
    
   <!-- Botón para confirmar la alarma -->  
   <Button  
   android:id="@+id/btnSetAlarm"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:text="Confirmar Alarma"  
   android:layout\_gravity="center"/>  
   <Button  
   android:id="@+id/btnCancelarAlarma"  
   android:layout\_width="wrap\_content"  
   android:layout\_height="wrap\_content"  
   android:backgroundTint="#F44336"  
   android:text="Cancelar"  
   android:layout\_gravity="center"/>  
    
  </LinearLayout>

# 5. Burndown Chart

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SPRINT BURNDOWN CHART | | | | | | | | | | | | |
|  |
| Backlog ID | Historia de Usuario | INICIAL ESTIMADO | 11-Abr | 12-Abr | 13-Abr | 14-Abr | 15-Abr | 16-Abr | 17-Abr | 18-Abr | 19-Abr | 20-Abr |  |
| DIA 0 | DIA 1 | DIA 2 | DIA 3 | DIA 4 | DIA 5 | DIA 6 | DIA 7 | DIA 8 | DIA 9 | DIA 10 |  |
| 1.2.6 | Historia de usuario 2.1 Inicio de sesión | 5 | 2 | 2 | 1 |  |  |  |  |  |  |  |  |
| 2.1.1 | Historia de Usuario 2.1 Registro y Gestion de rutina | 5 | 2 | 1 | 1 | 1 |  |  |  |  |  |  |  |
| 2.1.2 | 6 |  |  | 1 | 1 | 1 | 1 | 1 | 1 |  |  |  |
| 2.1.3 | 4 |  |  | 1 | 1 |  |  |  |  |  | 2 |  |
| 2.1.4 | 6 |  | 1 |  | 1 |  |  | 1 | 1 | 2 |  |  |
| 2.1.5 | 6 |  |  |  | 2 | 1 | 1 | 1 | 1 |  |  |  |
| 2.2.1 | Historia de Usuario 2.2 Modificiación de rutina | 4 |  |  |  |  |  |  | 1 | 1 | 1 | 1 |  |
| 2.2.2 | 4 | 2 | 2 |  |  |  |  |  |  |  |  |  |
| 2.2.3 | 4 |  |  |  |  | 1 | 1 | 1 | 1 |  |  |  |
| 2.2.4 | 6 |  |  |  |  |  | 1 | 1 | 1 | 1 | 2 |  |
| 2.2.5 | 6 |  |  |  |  | 1 | 1 | 1 | 1 | 1 | 1 |  |
| 1.3.1 | Historia de usuario 3.1 Configuración y Gestión de alarma | 5 |  |  |  |  |  | 1 | 1 | 1 |  | 2 |  |
| Tiempo Trabajado | | 61 | 55 | 49 | 45 | 39 | 35 | 29 | 21 | 13 | 8 | 0 |  |
| Tiempo Estimado | | 61 | 55 | 49 | 43 | 37 | 31 | 24 | 18 | 12 | 6 | 0 |  |

# 6. Análisis del Sprint

# Durante el Sprint 02, se espera que el equipo mantenga un desempeño coordinado en el diseño e implementación de las interfaces de rutina y configuración de alarmas, reforzando la colaboración entre diseño y desarrollo frontend. Se anticipa que uno de los desafíos principales estará en la implementación de funcionalidades offline, especialmente relacionadas con el almacenamiento local de las rutinas diarias y la modificación de estas. Además, se prestará especial atención a las validaciones de tiempo y la lógica para cancelar cambios, lo que podría requerir pruebas más exhaustivas. Con base en lo aprendido en el sprint anterior, se priorizará una planificación más detallada, una revisión técnica temprana de las dependencias y una mejor definición de casos de prueba para evitar bloqueos durante el desarrollo.

# 7. Conclusiones

El Sprint 02 permitirá consolidar la funcionalidad central de la aplicación relacionada con la gestión personalizada de la rutina diaria del usuario. Se buscará avanzar significativamente en el diseño, implementación y validación de los formularios para registrar, editar y modificar la rutina, así como en la configuración inicial de alarmas personalizadas.