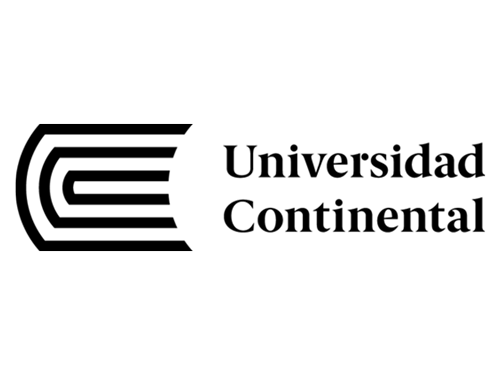
***"Año de la recuperación y consolidación de la economía peruana"***



**FACULTAD DE INGENIERÍA**

**TEMA:**

**TP2.E10 Manual Tecnico**

**Curso:**

**Taller de proyectos 2**

**NRC: 17185**

**Docente:**

**Americo Estrada Sanchez**

**Alumno:**

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

**CUSCO -  PERÚ**

**2025**

Manual Técnico — SleepApp

**Fecha**: 18/06/2025

**Autor**: Christian Castro Usto

**Versión**: 1.0

# 1. Introducción

Este manual técnico documenta los aspectos internos del desarrollo de **SleepApp**, una aplicación móvil desarrollada en Android para la gestión personalizada de la rutina diaria y la mejora del sueño. El documento está dirigido a desarrolladores, técnicos de soporte y profesionales interesados en comprender la **estructura, arquitectura, tecnologías utilizadas y el funcionamiento interno** del sistema.

SleepApp tiene como objetivo proporcionar una experiencia integral al usuario, permitiéndole registrar su rutina, configurar alarmas personalizadas, visualizar estadísticas de sueño y acceder a contenido de relajación, todo almacenado localmente mediante SQLite. Este manual ofrece una guía técnica completa para facilitar el mantenimiento, la evolución del sistema y su escalabilidad en futuras versiones.

.

# 2. Tecnologías Utilizadas

|  |  |
| --- | --- |
| Componente | Tecnología |
| Frontend (Interfaz) | Android SDK (Java) + XML (Material Design) |
| Lógica de Negocio | Java (Clases de control y validación) |
| Base de Datos | SQLite (almacenamiento local) |
| Persistencia de Datos | SQLite (sincronización local/offline) |
| Reproducción de Medios | Android MediaPlayer (videos de relajación) |
| Notificaciones/Alarmas | AlarmManager + NotificationManager de Android |

# 3. Estructura del Proyecto

# SleepApp/

# │

# ├── manifests/

# │ └── AndroidManifest.xml

# │

# ├── java/

# │ └── com.example.sleepapp/

# │ ├── MainActivity.java ← App.js

# │ ├── MenuPrincipal.java ← Pantalla de inicio

# │ ├── Registrar.java ← Registro.js

# │ ├── LoginActivity.java ← Login.js

# │ ├── RegistrarAlarma.java ← NuevaTarea.js

# │ ├── NotificacionesActivity.java ← Notificaciones

# │ ├── EstadisticasActivity.java ← Vista de estadísticas

# │ ├── GraficoBarras.java ← Gráfico barras

# │ ├── GraficoCircular.java ← Gráfico circular

# │ ├── AlarmReceiver.java ← Alarmas automáticas

# │ ├── ApagarAlarma.java ← Detener alarmas

# │ ├── Usuario.java ← Modelo de usuario

# │ ├── SqlLiteHelper.java ← Configuración de base de datos

# │ └── ... otros componentes

# │

# ├── res/

# │ ├── layout/

# │ │ ├── activity\_apagar\_alarma.xml

# │ │ └── activity\_estadisticas.xml

# │ │ └── activity\_listardatos.

# │ │ └── activity\_login.xml

# │ │ └── activity\_main.xml

# │ │ └── activity\_menu\_principal.xml

# │ │ └── activity\_registrar.xml

# │ │ └── activity\_registrar\_alarma.xml

# │ │ └── activity\_registrar\_rutina.xml

# │ └── drawable/

# │ └── ... imágenes, íconos

# │

# ├── test/

# │ └── com.example.sleepapp/

# │

# └── androidTest/

# └── com.example.sleepapp/

# 4. Arquitectura General

Arquitectura en **capas**:

* **Capa de Presentación:** Android (Java) + XML (Material Design) para la interfaz de usuario móvil.
* **Capa de Negocio:** Lógica implementada en Java dentro de las Activities y clases de control para el procesamiento de datos, cálculos de sueño y validaciones.
* **Capa de Modelo:** Clases Java que representan las entidades del sistema (Usuario, Rutina, Alarma, etc.), enlazadas a la base de datos SQLite.
* **Capa de Persistencia:** SQLite como base de datos local, con acceso a través de DAOs (UsuarioDao, RutinaDao, AlarmaDao) y gestionada mediante DatabaseHelper.

# 5. Descripción de Componentes

### 5.1 Frontend (Interfaz Móvil Android)

* **Lenguaje:** Java para Android (usando Android Studio).
* **Estilos:** XML con Material Design.
* **Comunicación con base de datos:** Mediante clases Java y DAOs conectadas a SQLite.
* **Componentes clave (Activities):**
  + Registrar.java: Formulario de registro de nuevos usuarios.
  + LoginActivity.java: Autenticación e inicio de sesión local.
  + RegistroRutina.java: Ingreso y edición de la rutina diaria del usuario.
  + RegistrarAlarma.java: Configuración de alarmas personalizadas.
  + VideosActivity.java: Reproducción de videos de relajación.
  + EstadisticasActivity.java: Visualización de datos estadísticos (horas dormidas, promedios).

### 5.2 Backend (Lógica de Negocio integrada en la app)

* **Tecnología:** Java (en las mismas Activities y clases auxiliares).
* **Gestión de operaciones CRUD:** Se realiza dentro de DatabaseHelper.java.
* **Validaciones:** Lógica incorporada para controlar:
  + Registro e inicio de sesión (verificación de datos).
  + Reglas para ingreso/modificación de rutinas y alarmas.
  + Validación de campos (tiempos, estados, formatos).
* **Funcionalidad offline:** Toda la aplicación funciona sin conexión, almacenando datos en SQLite local.

### 5.3 Base de Datos

* **Motor:** SQLite (local, embebido en la app).
* **Gestión de acceso:** Implementado mediante DAOs y DatabaseHelper.java.
* **Tablas del esquema:**
  + usuarios(id\_usuario, dni, nombre, apellidos, correo, contraseña, fecha\_nacimiento, fecha\_registro)
  + rutinas(id\_rutina, edad, hora\_trabajo, hora\_ejercicios, hora\_actividades\_recreativas, cantidad\_estimulantes, horas\_pantalla, cantidad\_estres, id\_usuario)
  + alarmas(id\_alarma, hora, tono, estado, id\_usuario)
  + relajacion(id\_video, titulo, url\_video, duracion, id\_usuario)
  + estadisticas(id\_estadistica, dia, horas\_dormidas, promedio, id\_rutina)

## 6. Instalación y Ejecución

## 6.1 Requisitos

 **Android Studio** (versión recomendada: Electric Eel o superior)

 **Java Development Kit (JDK) 8 o superior**

 **Dispositivo físico con Android 8.0+** o **emulador configurado**

 **Git** (para clonar el repositorio si es necesario)

## 6.2 Backend

## git clone https://github.com/ChrisCU16/SLEEPAPP\_TP2.git

## cd SLEEPAPP\_TP2

## 6.3 Abrir y ejecutar el proyecto en Android Studio

 Abrir Android Studio.

 Seleccionar "Open an existing project" y elegir la carpeta SLEEPAPP\_TP2.

 Esperar a que se sincronicen las dependencias Gradle.

 Conectar un dispositivo Android físico o iniciar un emulador.

 Hacer clic en **Run (▶)** para compilar y ejecutar la app.

## 6.4 Base de Datos

 No **requiere configuración externa.**

 La base de datos SQLite se crea automáticamente al ejecutar la aplicación por primera vez mediante la clase DatabaseHelper.java.

 Las tablas incluyen: usuarios, rutinas, alarmas, relajacion y estadisticas.

 Todos los datos se almacenan localmente en el dispositivo.

# 7. Seguridad

 Contraseñas **almacenadas de forma segura** utilizando funciones hash (como MessageDigest en Java).

 Validación **de entradas de usuario** en los formularios de registro e inicio de sesión.

 Restricción **de acceso a datos**: solo el usuario autenticado puede acceder a sus propias rutinas, alarmas o estadísticas.

 Protección **local**: uso de SharedPreferences para almacenar información mínima, evitando datos sensibles sin cifrado.

 En **futuras versiones**, se recomienda integrar autenticación segura en la nube (Firebase/Auth).

# 8. Buenas Prácticas

* **Código estructurado por capas (MVVM)**: separación clara entre interfaz, lógica y datos.
* **Modularización de funcionalidades**: clases independientes para cada módulo (Usuario, Rutina, Alarma, etc.).
* **Uso de constantes y recursos centralizados** para colores, strings y estilos.
* **Tratamiento de errores y validaciones** implementado antes de ejecutar operaciones críticas.
* **Persistencia local con SQLite**, con clases DAO bien definidas.

# 9. Futuras Mejoras

* **Sincronización en la nube con Firebase** para acceso desde múltiples dispositivos.
* **Funcionalidad de edición y eliminación** para rutinas y alarmas.
* **Notificaciones inteligentes** para mejorar la adherencia al sueño saludable.
* **Panel gráfico interactivo** para mostrar tendencias y estadísticas personalizadas.
* **Integración con sensores de sueño o pulseras inteligentes** para análisis más precisos.

# 10. Contacto

Para soporte técnico, sugerencias o colaboración, puedes escribir a:  
📧 **sleepapp.dev@proyecto.edu.pe**