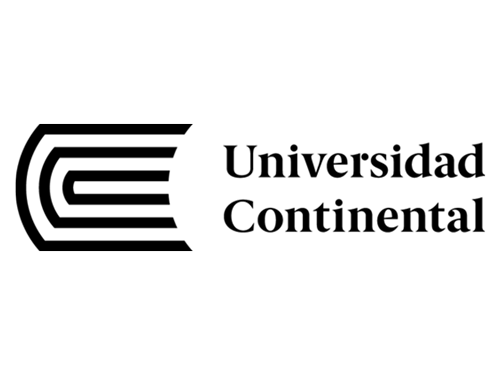
***"Año de la recuperación y consolidación de la economía peruana"***



**FACULTAD DE INGENIERÍA**

**TEMA:**

**TP2.E14 Manual De Pruebas**

**Curso:**

**Taller de proyectos 2**

**NRC: 17185**

**Docente:**

**Americo Estrada Sanchez**

**Alumno:**

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

**CUSCO -  PERÚ**

**2025**

Manual de Pruebas — SLEEPAPP

**Fecha**: 02/07/2025

**Autor**: Christian Castro Usto

**Versión**: 1.0

# 1. Introducción

Este manual describe el proceso de verificación y validación de la aplicación **SleepApp**, incluyendo los tipos de pruebas realizadas, los casos de prueba definidos, herramientas utilizadas y criterios de aceptación establecidos. Está dirigido al equipo de aseguramiento de la calidad (QA) y a los desarrolladores responsables de validar que el sistema funcione correctamente antes de su despliegue.

.

# 2. Objetivos

* Verificar que cada funcionalidad cumpla con los requisitos funcionales y no funcionales definidos.
* Identificar errores antes del lanzamiento y corregirlos a tiempo.
* Asegurar que la aplicación sea estable, usable y segura en diferentes entornos de ejecución.

# 3. Tipos de Pruebas Aplicadas

**3.1 Pruebas Unitarias**

* Validan clases y funciones individuales del backend (SQLite, DAO) y del frontend (actividades Android).
* Herramientas: **JUnit**, **Mockito**.

**3.2 Pruebas de Integración**

* Validan la interacción entre componentes como la vista (Activity) y la base de datos SQLite.

**3.3 Pruebas Funcionales**

* Verifican funcionalidades clave como registro, ingreso de rutina, configuración de alarmas, y reproducción de videos.

**3.4 Pruebas de Regresión**

* Se ejecutan al finalizar cada sprint para asegurar que nuevas funcionalidades no rompan las anteriores..

**3.5 Pruebas Manuales (Caja Negra)**

* Realizadas desde la perspectiva del usuario final usando dispositivos Android reales.
* Se valida la experiencia completa sin conocer la lógica interna.

# 4. Ambiente de Pruebas

|  |  |
| --- | --- |
| Componente | Versión |
| Android Studio | Flamingo+ |
| Java SDK | 11 |
| SQLite | nativo Android |
| Dispositivo | Android 10–13 |
| Emulator | Pixel 4 (API 30) |

# 5. Casos de Prueba

**5.1 Registro de Usuario**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Descripción | Entrada | Resultado Esperado | Estado |
| TC01 | Registro exitoso | Correo, contraseña válidos | Redirección al login | Aceptado |
| TC02 | Correo ya registrado | Correo existente | Mensaje de "Usuario ya registrado" | Aceptado |
| TC03 | Campos vacíos | Ninguno | Mensaje de "Campos requeridos" | Aceptado |

**5.2 Inicio de Sesión**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Descripción | Entrada | Resultado Esperado | Estado |
| TC04 | Login exitoso | Credenciales válidas | Acceso al menú principal | Aceptado |
| TC05 | Login fallido | Contraseña incorrecta | Mensaje de error de autenticación | Aceptado |

**5.3 Gestión de Rutinas**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Descripción | Entrada | Resultado Esperado | Estado |
| TC06 | Registrar rutina diaria | Horarios de estudio, trabajo, etc. | Datos guardados y visibles | Aceptado |
| TC07 | Validar tiempos inconsistentes | Horario de descanso negativo | Mensaje de validación | Aceptado |
| TC08 | Modificar rutina | Cambios aplicados | Rutina actualizada correctamente | Aceptado |

# 5.4 Alarmas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Descripción | Entrada | Resultado Esperado | Estado |
| TC09 | Configurar nueva alarma | Hora, tono | Alarma almacenada en local | Aceptado |
| TC10 | Funcionamiento offline | Sin internet | Alarma sigue funcionando | Aceptado |

# 6. Resultados

Todas las pruebas realizadas al finalizar el Sprint 2 y durante el Sprint 3 han sido satisfactorias. Las funcionalidades principales como el acceso, registro de rutina y configuración de alarmas están operativas y estables en dispositivos de prueba.

# 7. Criterios de Aceptación

* El sistema permite registrar, editar y visualizar rutinas diarias.
* Las alarmas personalizadas funcionan correctamente, incluso sin conexión.
* El acceso a la aplicación (login/registro) funciona con validaciones.
* El sistema no presenta errores críticos durante la navegación normal.

# 8. Herramientas Utilizadas

* **Android Studio**: entorno principal de pruebas.
* **JUnit**: pruebas unitarias de clases lógicas y controladores.
* **Mockito**: pruebas con objetos simulados (mocked DAOs).
* **ADB + Logcat**: para detección de errores en ejecución.
* **Dispositivos físicos + emulador**: pruebas manuales.

# 9. Recomendaciones Futuras

* Integrar pruebas de interfaz automatizadas con **Espresso**.
* Implementar pruebas E2E con múltiples flujos de usuario.
* Validar comportamiento en dispositivos con batería optimizada y modo oscuro.
* Agregar testing para cambios en configuración del sistema (idioma, brillo, etc.).

# 10. Contacto QA

Para reportes de errores o coordinación de pruebas:

📧 73024009@continental.edu.pe

📞 +51 966384355