



# **Informe de Eventos de Implementación Piloto**

**Curso:** 4° Medio A — T.P. Programación  
**Proyecto:** Aplicación de Escritorio “Gestor de Tareas”  
**Alumno(a):** Jorge Mesias  
**Fecha:** 06/11/2025

# Resumen

El presente informe expone el proceso de implementación piloto de la aplicación de escritorio “**Gestor de Tareas**”, desarrollada por el curso 4º Medio A de la especialidad de Programación. Durante esta etapa se evaluó el desempeño funcional del sistema, la experiencia de usuario y la eficiencia del código. Los resultados reflejaron una alta usabilidad, con pequeños ajustes pendientes en el manejo de base de datos. Se concluye que la aplicación cumple satisfactoriamente los objetivos propuestos y se recomienda continuar su optimización mediante mejoras en la interfaz y la seguridad de datos.

## 1. Introducción

El propósito de este informe es presentar los resultados de la implementación piloto de la aplicación de escritorio “**Gestor de Tareas**”, diseñada como parte del módulo de desarrollo de aplicaciones. Este proyecto tuvo como objetivo aplicar conocimientos de programación estructurada y orientada a objetos en un entorno de desarrollo real.

La aplicación permite al usuario gestionar y organizar tareas personales, registrar actividades pendientes y asignarles fechas límite. El informe detalla las etapas de desarrollo, tecnologías utilizadas, resultados obtenidos y recomendaciones finales derivadas del proceso de implementación.

El proyecto se llevó a cabo con el objetivo de fortalecer las competencias técnicas del alumnado y comprobar el correcto funcionamiento del software en un entorno controlado de usuarios finales.

## 2. Descripción y Alcance de la App

### - Descripción de la Aplicación

- **Nombre:** Gestor de Tareas
- **Propósito:** Facilitar la organización personal mediante la creación, edición y eliminación de tareas.

- **Funcionalidades:**
  - Crear y eliminar tareas
  - Marcar tareas como completadas
  - Filtrar por fecha o estado
  - Guardado automático en base de datos
- **Público Objetivo:** Estudiantes y profesionales que requieran una herramienta de gestión personal simple y funcional.

#### - Alcance de la Implementación Piloto

- **Objetivos:** Evaluar la funcionalidad general, rendimiento y satisfacción del usuario.
- **Ubicación y Duración:** Laboratorio de informática del establecimiento; duración de una semana.
- **Usuarios Involucrados:** 10 estudiantes del curso y el docente guía.
- **Nivel de Satisfacción:** 85% de los usuarios calificaron la experiencia como “muy buena”.
- **Consideraciones Adicionales:**
  - **Desafíos:** Manejo de errores en conexión a la base de datos.
  - **Lecciones Aprendidas:** Importancia de validar entradas y realizar pruebas unitarias antes de la implementación.

### 3. Codificación y Tecnologías Utilizadas

#### - Codificación y Desarrollo

- **Arquitectura:** Estructura de tres capas (interfaz, lógica y datos).
- **Lenguaje:** Python (paradigma orientado a objetos).
- **Base de Datos:** SQLite.
- **Metodología de Trabajo:** Metodología ágil tipo *Scrum*, con reuniones diarias y planificación semanal.

## - Tecnologías Utilizadas

- **IDE:** Visual Studio Code
- **Bibliotecas y APIs:** Tkinter (interfaz gráfica), SQLite3 (gestión de base de datos), hashlib (hashing).
- **Pruebas:** Testing manual y validación por pares.
- **Seguridad:** Aplicación de hash MD5 en contraseñas almacenadas.

## 4. Resultados y Conclusiones

### - Resultados

- **Desempeño de la Aplicación:** Funcionó correctamente en todas las pruebas funcionales.
- **Usabilidad y Experiencia de Usuario:** Interfaz clara y sencilla; se recomienda agregar modo oscuro.
- **Problemas Técnicos:**
  - Error esporádico al iniciar sesión por validación de hash.
  - Leve lentitud al cargar grandes volúmenes de tareas.

### - Conclusiones

- **Logros:** Implementación exitosa del CRUD de tareas y autenticación de usuarios.
- **Lecciones Aprendidas:** Importancia de planificar estructuras de base de datos eficientes.
- **Recomendaciones Finales:**
  - Optimizar consultas SQL.
  - Implementar un sistema de notificaciones.
  - Realizar pruebas automatizadas en futuras versiones.

## 5. Análisis y Recomendaciones

### - Análisis

El análisis de la implementación refleja un cumplimiento del 90% de los objetivos planteados. Los usuarios valoran la funcionalidad principal y el diseño intuitivo, aunque se detectó la necesidad de optimizar tiempos de respuesta y mejorar la persistencia de datos. Se observan patrones de satisfacción alta en tareas básicas y menores incidencias en funciones secundarias.

### - Recomendaciones

- **Mejoras Funcionales (Backend):** Refinar consultas SQL y agregar logs de errores.
- **Optimización de Usabilidad (Frontend):** Incluir temas visuales y mensajes de retroalimentación visual.
- **Plan de Acción:**
  1. Revisión del modelo de datos.
  2. Implementar pruebas automáticas.
  3. Desarrollar actualización de interfaz.
  4. Evaluar rendimiento post-optimización.

## 6. Antecedentes Bibliográficos

- Python Software Foundation. *Documentación oficial de Python 3*.
- Tkinter Documentation. *Python Interface for Tk GUI Toolkit*.
- SQLite. *Database Engine Documentation*.
- Sommerville, I. (2016). *Ingeniería de Software* (10<sup>a</sup> edición). Pearson Education.