

FACULTAD COMUNITARIA CAACUPÉ – ING. INFORMÁTICA



Proyecto Lenguajes Visuales II

Adivinar un Número del 1 al 100

Materia: Lenguaje Visuales II

Tutor: Ing. Ricardo Maidana

Autor: Nayeli Luján León Velázquez

Curso: 2°

Semestre: 4°

Turno: Mañana

Sección: “B”

Caacupé-Paraguay

2025.

1. Introducción

El presente proyecto consiste en el desarrollo de una aplicación de escritorio para la gestión de libros, implementada en el lenguaje de programación Python (versión 3.x), utilizando la biblioteca Tkinter para la construcción de la interfaz gráfica de usuario (GUI). Esta propuesta surge como trabajo integrador de la asignatura Lenguajes Visuales II, con el objetivo de aplicar de manera práctica los conocimientos adquiridos a lo largo del curso, incluyendo programación estructurada, manejo de eventos, diseño visual, validaciones, persistencia de datos y organización modular del código.

La aplicación está orientada a facilitar el registro, edición, eliminación y visualización de libros en una biblioteca personal o institucional, mediante una interfaz intuitiva y funcional. La información se almacena de forma persistente en un archivo CSV, lo que permite conservar los datos entre sesiones y mantener un respaldo accesible. Además, se han incorporado controles de validación que aseguran la integridad de los datos ingresados, como la verificación de campos obligatorios y la confirmación de acciones sensibles como la eliminación de registros.

Este proyecto no solo representa una solución técnica funcional, sino también una oportunidad para reflexionar sobre la importancia del diseño visual, la experiencia del usuario y la calidad del código en el desarrollo de software.

2. Objetivos del proyecto

Objetivo general: Desarrollar un sistema de gestión de biblioteca que permita administrar libros de manera eficiente, segura y visualmente agradable, utilizando Python y Tkinter, integrando buenas prácticas de programación, diseño de interfaz y almacenamiento de datos.

Objetivos específicos: Implementar una interfaz gráfica clara, coherente y amigable, utilizando widgets de Tkinter como Label, Entry, Button, Treeview y Scrollbar.

Gestionar información de libros mediante operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre un archivo CSV codificado en UTF-8.

Aplicar validaciones de entrada que garanticen la integridad de los datos, evitando errores comunes como campos vacíos o tipos de datos incorrectos.

Organizar el código de manera modular, separando funciones de gestión de datos, lógica de interfaz y eventos de usuario, facilitando el mantenimiento y la escalabilidad.

Proporcionar funcionalidades de navegación intuitiva y visualización ordenada de los registros, mejorando la experiencia del usuario.

3. Justificación técnica y pedagógica

El desarrollo de esta aplicación responde a una doble necesidad: por un lado, consolidar conocimientos técnicos adquiridos en la asignatura Lenguajes Visuales II; por otro, aplicar dichos conocimientos en un contexto práctico que simule una situación real de desarrollo de software. La elección de Python como lenguaje base se fundamenta en su sintaxis clara, su amplia comunidad y su versatilidad para proyectos educativos. Tkinter, como biblioteca gráfica nativa, permite construir interfaces funcionales sin necesidad de instalar dependencias externas, lo que facilita la portabilidad y el aprendizaje.

Desde el punto de vista pedagógico, el proyecto permite integrar conceptos clave como la programación modular, la validación de datos, el diseño visual y la persistencia de información. Además, promueve el pensamiento lógico, la organización del código y la mejora continua, habilidades esenciales para cualquier desarrollador en formación.

4. Público objetivo y contexto de uso

La aplicación está diseñada para usuarios que necesiten gestionar colecciones de libros de forma sencilla, sin requerir conocimientos avanzados en informática.

Esto incluye estudiantes, docentes, bibliotecarios escolares o cualquier persona que desee organizar sus lecturas personales. Su interfaz amigable y sus funcionalidades básicas permiten que el sistema sea utilizado en contextos educativos, bibliotecas pequeñas o como herramienta de apoyo en proyectos académicos.

El uso de un archivo CSV como sistema de almacenamiento garantiza que los datos puedan ser consultados, editados o exportados fácilmente, incluso sin abrir la aplicación, lo que amplía su utilidad en entornos donde se requiere flexibilidad y accesibilidad.

5. Metodología de desarrollo

El proyecto fue desarrollado siguiendo una metodología iterativa, basada en ciclos de mejora continua. En cada etapa se definieron objetivos específicos, se implementaron funcionalidades y se realizaron pruebas para validar el comportamiento del sistema. Esta forma de trabajo permitió detectar errores tempranos, ajustar el diseño visual y optimizar la experiencia del usuario.

Se priorizó la separación de responsabilidades en el código, dividiendo las funciones en bloques temáticos: gestión de datos, interfaz gráfica y eventos de usuario. Esta organización modular facilita el mantenimiento, la comprensión del código y la incorporación de futuras mejoras. Además, se documentaron internamente las funciones mediante comentarios explicativos, lo que refuerza la claridad y el profesionalismo del desarrollo.

6. Descripción de la estructura y funcionalidades

6.1 Estructura del código Configuración inicial y constantes.

Color de fondo uniforme (#e6ffe6) para coherencia visual.

Dimensiones mínimas y fuentes legibles para accesibilidad.

Funciones de gestión de datos

- `inicializar_csv()`: Crea el archivo libros.csv si no existe.
- `cargar libros()`: Lee los registros del archivo CSV.
- `guardar libro()`: Agrega un nuevo libro al archivo.
- Ventana de gestión de libros
- Campos de entrada para Título, Autor, Género y Año.
- Botones para Guardar, Editar, Eliminar y Refrescar.
- Tabla Treeview con Scrollbar vertical.

Funciones auxiliares

actualizar_tabla(), limpiar_campos(), validar_campos(), seleccionar_libro(), editar(), eliminar().

Menú principal

Ventana de bienvenida con botones para gestionar libros o salir.

6.2 Funcionalidades principales Agregar, editar y eliminar libros.

- Visualización en tabla con scroll.
- Validación de campos obligatorios y tipo de dato.
- Persistencia de datos en archivo CSV codificado en UTF-8.

7. Diseño de interfaz Colores y estética:

- ❖ Fondo verde menta, botones diferenciados por acción.
- ❖ Organización de widgets: Elementos alineados y centrados.
- ❖ Tabla de datos: Treeview con columnas claras y scroll vertical.
- ❖ Este diseño busca ofrecer una experiencia visual agradable, accesible y funcional, alineada con los principios de usabilidad aprendidos en clase.

8. Ventajas y fortalezas Interfaz sencilla y fácil de usar.

- ❖ Operaciones CRUD integradas en un solo entorno.
- ❖ Validaciones que aseguran calidad de datos.
- ❖ Código modular y documentado.
- ❖ Almacenamiento persistente y accesible.

9. Posibles mejoras Búsqueda y filtrado por título, autor o género.

- ❖ Ordenamiento por columnas en la tabla.
- ❖ Estilos personalizados con ttk.Style.
- ❖ Respaldo automático del archivo CSV.
- ❖ Migración a SQLite para mayor escalabilidad.
- ❖ Exportación a Excel o PDF para reportes.

10. Pruebas y validación Se realizaron pruebas funcionales para cada operación (agregar, editar, eliminar, visualizar), así como pruebas de validación para evitar errores por campos vacíos o datos incorrectos. La tabla se actualiza dinámicamente y se verificó que los datos se guardan correctamente en el archivo CSV. El sistema responde de forma estable y confiable ante diferentes escenarios de uso.

11. Documentación técnica El código fuente incluye comentarios explicativos en cada función, lo que facilita su comprensión y mantenimiento. Se elaboró además un documento complementario con instrucciones de uso,

estructura del sistema y posibles mejoras, pensado para acompañar la defensa oral y servir como guía para futuros desarrolladores.

12. Reflexión personal Este proyecto representó un desafío enriquecedor que permitió consolidar conocimientos técnicos y desarrollar habilidades de diseño visual. A lo largo del proceso, se aprendió a resolver problemas, mejorar la estética de la interfaz y pensar en el usuario final. La experiencia fue motivadora y dejó una base sólida para futuros desarrollos más complejos.

13. Conclusión

El proyecto de Gestor de Biblioteca en Python con Tkinter cumple con los objetivos planteados en la asignatura Lenguajes Visuales II, demostrando competencias en:

Diseño de interfaces gráficas funcionales y amigables.

Aplicación de programación estructurada y modular.

Gestión de datos persistentes y validación de información.

Integración de funcionalidades completas de CRUD en un solo programa.

Además, representa una base sólida para futuros desarrollos, permitiendo incorporar mejoras visuales, optimización de datos y nuevas funcionalidades. Este trabajo refleja el compromiso con la calidad, la estética y la funcionalidad, y constituye una muestra del crecimiento técnico alcanzado durante el curso.