

Proyecto de aplicación 1 2025s2 Biblioteca Digital

Objetivo General

Desarrollar un simulador en consola que lea y procese registros de préstamos de libros desde un archivo texto implementando estructura de datos nativa para generar reportes informativos de una biblioteca digital.

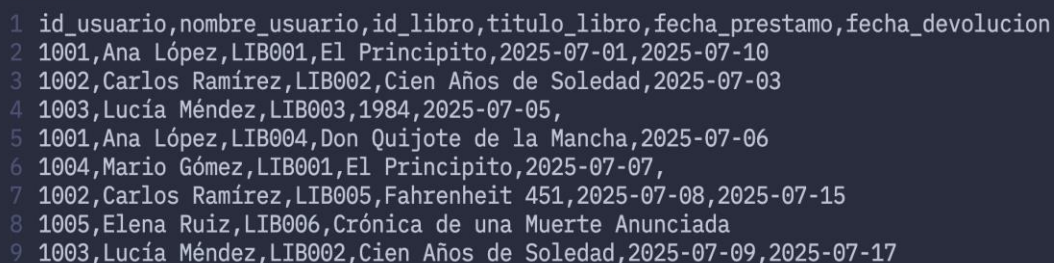
Objetivos Específicos

- Leer y realizar análisis léxico básico de archivos de texto con información estructurada.
- Manipular estructuras de datos para almacenar y organizar información.
- Generar reportes en consola y exportar resultados en formato HTML.
- Consolidar estadísticas claves del sistema como libros y usuarios frecuentes.

Enunciado

La URL le solicita que desarrolle un programa que funcione desde la consola para simular el sistema de gestión de préstamos de una biblioteca digital. El sistema deberá de ser capaz de procesar un archivo de texto que contiene múltiples registros de préstamos de libros, analizar la información y generar reportes presentando la información de manera clara.

Archivo de préstamos (.lfa)



```
1 id_usuario,nombre_usuario,id_libro,titulo_libro,fecha_prestamo,fecha_devolucion
2 1001,Ana López,LIB001,El Principito,2025-07-01,2025-07-10
3 1002,Carlos Ramírez,LIB002,Cien Años de Soledad,2025-07-03
4 1003,Lucía Méndez,LIB003,1984,2025-07-05,
5 1001,Ana López,LIB004,Don Quijote de la Mancha,2025-07-06
6 1004,Mario Gómez,LIB001,El Principito,2025-07-07,
7 1002,Carlos Ramírez,LIB005,Fahrenheit 451,2025-07-08,2025-07-15
8 1005,Elena Ruiz,LIB006,Crónica de una Muerte Anunciada
9 1003,Lucía Méndez,LIB002,Cien Años de Soledad,2025-07-09,2025-07-17
```

En donde:

- **id_usuario:** Identificador numérico único del usuario que realiza el préstamo.
- **nombre_usuario:** Nombre completo del usuario.

- **id_libro:** Código único que identifica el libro dentro del sistema.
- **titulo_libro:** Título del libro que fue prestado.
- **fecha_prestamo:** Fecha en que se realizó el préstamo en formato (YYYY-MM-DD).
- **fecha_devolucion:** Fecha en que se espera la devolución del libro, Si este vacío, no ha sido devuelto el libro, la fecha de devolución debe de esta en formato (YYYY- MM-DD).

Todos los prestamos deben de ser almacenados en memoria. **Todos los códigos en los préstamos deben existir en el catálogo correspondiente.**

Reportes solicitados:

1. Historial de Prestamos: Deberá generar una tabla con todos los préstamos registrados.
2. Listado de Usuarios: Deberá generar una tabla de **usuarios únicos** que han realizado préstamos.
3. Listado de Libros Prestados: Deberá generar una tabla con todos los libros que han sido prestados, **sin duplicados**.
4. Estadísticas de Préstamos: Deberá de generar una tabla que contenga el total de préstamos, el libro más prestado, el usuario más activo y el total de usuarios únicos.
5. Prestamos Vencidos: Deberá de generar una tabla con cuya fecha_devolucion ya ha pasado y aún no han sido devueltos.

IMPORTANTE: Todos los reportes deberán de ser generados para formato HTML.

Menú

Se debe implementar un menú principal que contenga al menos las siguientes opciones:

- Cargar usuarios
- Cargar libros
- Cargar registro de préstamos desde archivo
- Mostrar historial de préstamos
- Mostrar listado de usuarios únicos
- Mostrar listado de libros prestados
- Mostrar estadísticas de préstamos
- Mostrar préstamos vencidos

- Exportar todos los reportes a HTML
- Salir

Toda la aplicación deberá ser desarrollada en modo consola. Si la aplicación encuentra algún carácter que no corresponde al alfabeto, o al patrón del dato, **deberá indicar el número de línea, la posición y el carácter erróneo, sin detener la ejecución.** Iniciando la lectura en la siguiente línea del archivo.

Entregables

- Link al repositorio de GitHub en el apartado de Proyecto 1 habilitado en el portal
- En el repositorio de GitHub debe de encontrarse lo siguiente:
 - Manual de usuario
 - Manual técnico
 - Código fuente
- Fecha de presentación del proyecto: **8 de septiembre en periodo de clase.**

Criterios de evaluación

- Calidad de la documentación: ortografía, orden, limpieza y que esté completa.
- Claridad y modularidad del código.
- Funcionalidad: DEBE cumplir a cabalidad con todos los requerimientos.
- Manejo adecuado de errores y excepciones.

Criterio	Descripción	Valor (%)	Excelente (100%)	Bueno (80%)	Regular (60%)	Deficiente (30%)	No se evidencia (0%)
1. Funcionalidad y cumplimiento de requisitos	El programa implementa correctamente todas las funciones solicitadas: carga de datos, reportes, manejo de errores,	35	Cumple todos los requerimientos sin fallos	Cumple 280% de funcionalidades, con errores menores	Cumple 260% con errores que afectan usabilidad	Cumple solo 230% con muchas fallas y errores	Implementación fallida
2. Manejo de errores y análisis léxico básico	Valida correctamente entradas, identifica caracteres fuera de alfabeto e informa	15	Manejo robusto de errores léxicos y validación	Identifica la mayoría de errores léxicos y validación	Manejo parcial de errores léxicos y validación	Identifica sólo algunos de los errores léxicos	No implementa análisis léxico como se solicitó
3. Diseño y codificación	Claridad, modularidad, uso de funciones nativas para la gestión de cadenas y estructuras.	10	Código modular, claro y documentado. Solo usa funciones nativas para gestionar cadenas y propias para las estructuras	Código modular, en su mayoría bien documentado. Uso adecuado de funciones nativas para cadenas y propias para estructuras; con errores menores	Código poco modular o confuso. Dificultades en uso de funciones nativas para cadenas y propias para estructuras	Código desordenado y sin modularidad. Documentación insuficiente o ausente. Uso inadecuado de funciones nativas y propias	No presenta código funcional o incumple requisitos de diseño
4. Documentación técnica y manual de usuario	Manual técnico con diagramas de flujo, clases y explicación de implementación; manual de usuario con instrucciones claras.	10	Documentación completa, incluye los diagramas, la documentación es clara y ordenada	Documentación casi completa con mínimos vacíos, falta alguno de los diagramas	Documentación parcial, estructura poco clara, diagramas incompletos	Documentación deficiente, no incluye diagramas	No presenta documentación técnica y de usuario
5. Evidencia de competencias CG2 – Liderazgo solidario y transformador	Trabajo colaborativo, coordinación y aporte equitativo.	5	Colaboración efectiva, roles claros y equitativos	Colaboración adecuada con pequeños desequilibrios	Colaboración desigual, no colabora de forma constante	Se evidencia una colaboración mínima	No se evidencia colaboración alguna
6. Evidencia de competencias CG4 – Gestión del conocimiento técnico-científico	Integración y aplicación de conceptos de análisis léxico, estructuras de datos y generación de reportes.	15	Explica y aplica fundamentos teóricos de forma apropiada en todo el proyecto	Aplica la mayoría de conceptos	Aplicación parcial de teoría	Se evidencia una aplicación mínima de la teoría	No se evidencia aplicación de los conceptos teóricos solicitados
7. Evidencia de competencias CE9 – Gestión estratégica de TI	Desarrollo de un prototipo funcional que organiza información y genera reportes para toma de decisiones.	10	Prototipo sólido con integración completa de funcionalidades	Prototipo funcional con fallas menores	Prototipo limitado o incompleto, con errores que afectan significativamente el uso o funcionamiento	Prototipo incompleto, con fallas y errores frecuentes	No se evidencia gestión adecuada de TI
8. Presentación y dominio individual	Defensa oral presencial: cada estudiante debe explicar, ejecutar y realizar cambios al código.	-100	(-0) Explica con claridad y demuestra dominio total	(-10) Explica bien con leves dificultades	(-30) Explicación incompleta, dominio parcial	(-60) Explicación deficiente, poco dominio del proyecto realizado	(-90) No demuestra comprensión de lo implementado

Consideraciones

- Este proyecto se trabajará en PAREJAS, las que ya formaron.
- La aplicación se debe desarrollar en modo consola.
- El manual técnico deberá contener por lo menos un diagrama de flujo y diagrama de clases para la solución.
- El análisis de la entrada de texto debe de realizarse manipulando la cadena **únicamente con funciones nativas del lenguaje**.
- Queda prohibida cualquier librería que facilite el análisis de la entrada de texto.

- El nombre del repositorio debe tener el siguiente formato LFA_2025S2_Equipo, dentro del repositorio deberá crear una carpeta llamada Proyecto1. Se debe de agregar como colaborador al docente del curso.
- **La calificación se realizará de forma PRESENCIAL.**
- Durante la calificación del proyecto es requerido que cada estudiante demuestre el dominio completo de lo implementado para que se le asigne la nota obtenida.
- Toda solución presentada debe compilar correctamente para poder tener derecho a revisión.
- IMPORTANTE: No puede usar librerías de terceros para implementar estructuras lineales.
- Cada entrega deberá hacerse en el espacio habilitado en el portal. Para la calificación se descargará el proyecto entregado por esta vía. No se aceptan entregas vía correo electrónico u otro medio.
- **Se podrá demandar que en la calificación presencial del proyecto se realicen cambios de funcionalidad.**
- **COPIA PARCIAL O TOTAL DEL PROYECTO TENDRÁ UNA NOTA DE 0 PUNTOS, Y SE NOTIFICARÁ A LA COORDINACIÓN DEL ÁREA PARA QUE SE APLIQUEN LAS SANCIONES CORRESPONDIENTES A LOS ALUMNOS DEL EQUIPO DE TRABAJO.**

Fecha de entrega: 05 de septiembre de 2025 antes de las 23:59, no se recibirán entregas después de la fecha y hora límite establecidas.