



NOMBRE DEL PROYECTO:

Nombre del Proyecto: Leaftime

Elevator Pitch (Descripción Rápida)

<https://github.com/itzamara15092000/ProyectoBiblioteca.git>

Es una herramienta de gestión y seguimiento de lectura desarrollada para lectores ávidos [Implied from code functionality: uploaded:Index.html]. La aplicación permite organizar el hábito de lectura rastreando el progreso de los libros (Pendiente, Leyendo, Terminado), ofreciendo estadísticas clave y calculando un ritmo de lectura diario sugerido para cumplir objetivos.

El Problema (Justificación / El "Porqué":

¿Qué problema real o necesidad están solucionando? La aplicación aborda la falta de un sistema centralizado y dinámico para que los lectores puedan:

1. Evitar la Pérdida de Seguimiento: Muchos lectores usan notas dispersas o la memoria para recordar dónde van o qué quieren leer, resultando en desorganización y olvido [Implied from code functionality: uploaded:Index.html].
2. Optimizar el Hábito: Hay una necesidad de convertir la lectura en un hábito medible y menos intimidante. La función de ritmo de lectura sugerido ayuda a establecer metas realistas y evita el estancamiento [Implied from code functionality: uploaded:Index.html].
3. Gestión de Inventario (TBR): Proporciona un inventario claro de libros "Por Leer" (TBR) y libros finalizados para una visión completa del historial y las metas [Implied from code functionality: uploaded:Index.html].



¿Quién usará esta aplicación?

Público Objetivo:

El público objetivo principal para la Agenda Literaria Personal está compuesto por individuos que buscan formalizar, organizar y mejorar su experiencia de lectura.

Segmentos Específicos

- Estudiantes Universitarios y Académicos: Que deben gestionar y rastrear la lectura de múltiples libros de texto, artículos o materiales de referencia simultáneamente.
- Lectores de Clubes de Lectura: Personas que necesitan asegurarse de cumplir con un plazo de lectura específico, ya que la aplicación ofrece un ritmo de lectura sugerido [Implied from code functionality: uploaded:Index.html].
- Lectores en Transición (de Físico a Digital): Usuarios que leen en múltiples formatos (Físico, eBook, Audiolibro) y necesitan consolidar su progreso en una sola herramienta [Implied from code functionality: uploaded:Index.html].

SECCIÓN B: EL ALCANCE (SCOPE)

Funcionalidades Principales (MVP)

- Gestión Completa del Inventario (CRUD): El usuario podrá añadir, editar y eliminar libros de su agenda, registrando atributos esenciales como título, autor, total de páginas y formato (Físico, eBook, Audiolibro).

- **Seguimiento de Progreso y Estado:** El sistema permite asignar un estado a cada libro (Pendiente, Leyendo, Terminado, Abandonado) y registrar el número de página actual para rastrear el progreso de la lectura activa.
- **Métricas de Ritmo de Lectura:** La aplicación calcula y sugiere un ritmo de lectura diario (páginas por día) y estima los días restantes para terminar un libro, basándose en la fecha de inicio y el progreso actual.
- **Panel de Estadísticas y Filtrado:** El dashboard mostrará un resumen de las estadísticas clave (Total Libros, Completados, Páginas Leídas, Leyendo Ahora) y permitirá filtrar la lista principal por el estado de lectura.
- **Persistencia de Datos (Local/Simulada):** La información de la agenda se guarda localmente (LocalStorage) y la aplicación simula la interacción con un backend SQL/API mediante una función asíncrona dedicada (saveBookAPI) para el guardado.

SECCIÓN C: LA TECNOLOGÍA (EL STACK)

El proyecto ya está usando E y N. La siguiente fase de desarrollo implicará la implementación de R y M para completar el Stack MERN.

| Componente | Objetivo (Stack MERN) | Implementación Actual (Prototipo) | Control de Versiones y Hospedaje |
|---------------|-----------------------------|---|---|
| Frontend | React (Interfaz de Usuario) | HTML, JavaScript Puro y Tailwind CSS. | Despliegue (FE): Vercel o Netlify (Para React/SPA). |
| Backend | Express.js (API REST) | Express.js sobre Node.js (Usado actualmente para el <i>mocking</i> de la API en saveBookAPI). | Despliegue (BE): Render o Railway (Para Node.js/Express). |
| Base de Datos | MongoDB (NoSQL) | LocalStorage (Almacenamiento temporal en el navegador). | Base de Datos: MongoDB Atlas (Nivel gratuito). |
| Control | Git / GitHub | Confirmado: Uso de Git como sistema de control y GitHub como <i>hosting</i> del repositorio | |

INTEGRANTES:

ITZAMARA SOTERO MARTINEZ

SANJUANA GARCÍA HERNÁNDEZ

YHAIRI XIMENA RIVERA REYES