

## AVANCE DE PROYECTO

# “Attor - Editorial de obras multiformato”

**Alumna:** Jocelyn Flores Gutiérrez

**Maestro:** Edgar Ivan Patricio Aizpuru

**Materia:** Desarrollo Full Stack

**Fecha:** 31/01/2026



## ● Índice de contenidos

---

● Índice de contenidos.....	2
● Idea del proyecto.....	3
● MVP (Producto Mínimo Viable).....	4
● Backend.....	5

---

En esta entrega deberás presentar el primer avance formal de tu proyecto final, combinando la idea general del sistema con un backend funcional mínimo (MVP), desarrollado únicamente con los conocimientos vistos hasta este punto del curso.

Este avance es conceptual y técnico y será la base sobre la cual se construirá el proyecto completo durante las siguientes semanas.

**Repositorio en GitHub:**

<https://github.com/JoFloresGutierrez06/Proyecto-Final---Desarrollo-Full-Stack>

**Documentación en Postman:**

<https://www.postman.com/jofloresgutierrez-602355/workspace/avance-de-proyecto-full-stack>

## ● Idea del proyecto

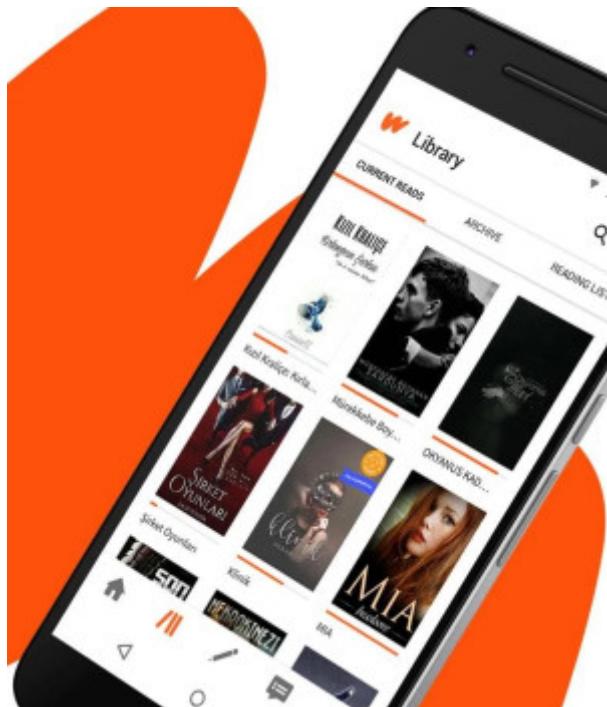
---

- *¿Qué quieres desarrollar?*
  - *¿Qué problema resuelve tu proyecto?*
  - *¿A quién va dirigido?*
  - *¿Cuál es el enfoque principal del sistema?*
- 

Se planea desarrollar una página web de una editorial de obras multiformato, que permita publicarlas y, así mismo, poder visualizar la colección de obras publicadas para poder leerlas.

Este proyecto va enfocado al problema que enfrentan algunos autores al tener que repartir el contenido de sus obras (texto, imágenes, cinematográficas, música, etc.) en distintas plataformas, porque no existe una opción práctica y popular para reunir todos los formatos de un proyecto en un mismo sitio y que los lectores lo puedan visualizar e interactuar con él.

Se pretende (posiblemente en un futuro) hacer una plataforma capaz de dividir a los usuarios en dos tipos de perfiles: lectores y autores. Siendo los **autores** capaces de tener las funciones de crear, editar, eliminar, publicar y ocultar obras que serán luego visualizadas por los **lectores** en una cuadrícula principal de portadas con título, siendo capaces de interactuar con las obras con un “me gusta” y guardándolas en su biblioteca personal.



## ● MVP (Producto Mínimo Viable)

---

- ¿Qué es lo mínimo que debe funcionar en tu proyecto?
  - ¿Qué funcionalidades incluye esta primera versión?
  - ¿Qué funcionalidades quedarán fuera por ahora?
- 

### Funciones mínimas del proyecto:

- Crear, actualizar, mostrar y eliminar obras.
- Página visible para visualizar las obras (publicadas y no publicadas) y botones para interactuar con las obras creadas.

### Funciones mínimas del primer entregable:

- Identificar una entidad para trabajar con ella. (Entidad elegida: obras)
- **Crear** obras especificando detalles como título, autor, descripción, portada y estado de publicación.
  - En el caso de la portada y publicación, si no se le especifica un valor, se le pone uno por default.
  - Los atributos de título, autor y descripción son obligatorios.
- **Mostrar** las obras publicadas, simulando lo que visualizará el lector.
- **Actualizar** los elementos de una obra en específico.
  - Los cambios pueden ser a un solo elemento o a varios al mismo tiempo.
- **Eliminar** obras por completo, independientemente de si se encuentran publicadas o no.

### Funciones planeadas para implementarse en un futuro:

- Añadir el atributo de “Género” a la entidad obras, para poder manejarlo en futuras implementaciones.
- Frontend que permita interactuar con las funcionalidades del backend.
- Visualizar las obras en una cuadrícula de portadas con título.
  - Al seleccionar una obra creada en específico:
    - Se **despliegan** más *detalles*.
    - Opción de **editar** los detalles iniciales.

## ● Backend

---

### **Entidad:** Obra

- Título (String)
- Autor (String)
- Descripción (String)
- Portada (String) - Se refiere al link de la imagen
- Publicado (Boolean)

---

**Publicado:** es un estado booleano que se refiere a la capacidad de la obra de estar al alcance (ser visible) para los lectores.

- Si tiene un valor (true) entonces podrá aparecer en la interfaz de los lectores.
  - Si tiene un valor (false) entonces permanecerá existente, pero no será accesible/visible para los lectores.
- 

- **GET - (Mostrar obras publicadas)**

El get con la dirección “/obras” muestra únicamente las obras publicadas (como se debería mostrar la interfaz de los lectores).

- **POST - (Crear obra)**

El post cuenta con dos tipos de atributos para la obra: los necesarios y los que, en caso de no especificar un valor, muestra uno por default.

**Atributos necesarios:**

- Título, Autor, Descripción

**Atributosopcionales:**

- Portada: (<http://portadadefault>)
- Publicado (false)

- **PUT - (Actualizar obra)**

El put está diseñado para que el usuario pueda mandar únicamente lo que le interesa cambiar. Por ejemplo, si únicamente le interesa cambiar el título y el estatus de publicado, entonces puede escribir esos sin necesidad de especificar los demás elementos.

- **DELETE - (Eliminar obra)**

El delete puede eliminar una obra, independientemente de si está publicada o no, y antes de hacerlo, verifica que el id proporcionado exista en el arreglo de obras.