

Proyecto Biblioteca

[Objetivo](#)

[Propuesta](#)

[Ingles vs Español](#)

[Casos de Uso](#)

[Básicos](#)

[Para explorar nuevas funcionalidades](#)

[Definiciones Base de Datos](#)

[Items](#)

[Copies](#)

[Members](#)

[Loans](#)

Objetivo

Vamos a realizar un proyecto lo suficientemente pequeño para aprender conceptos básicos y todo lo necesario para poder desarrollar proyectos que puedan crecer

- Diseñar una API
- Listar Entidades
- Ver Detalle Entidad
- Agregar Entidad
- Editar Entidad
- Borrar Entidad
- Relaciones 1 a N
- Relaciones N a N
- Trabajar con fechas
- Diferenciar
 - Controller: Pegamento endpoint con lógica de negocio
 - Service: Casos de Uso / Lógica de Negocio
 - Repository: Acceso a Fuente de Datos
 - Model: Representación de entidades
- Métodos DB por defecto de Hibernate (implementación de JPA)
- Métodos DB definibles por usuario sin SQL usando interfaz internet,
- Métodos DB usando SQL (problemático por poder depender de motor usado)

Propuesta

Vamos a realizar una API para gestionar una Biblioteca, esta biblioteca va a gestionar libros, discos, películas, juegos de mesa y cualquier otro contenido cultural que sea prestable.

Vamos a disponer de un catálogo.

Cada Elemento del Catálogo va a tener una serie de copias que harán referencia a las copias físicas que la biblioteca va a poder prestar.

La biblioteca tiene unos socios que van a poder hacer uso del servicio que presta la biblioteca, van a tener su ficha con datos de contacto.

Para poder dar el servicio lo único que nos queda es la gestión de Préstamos

Ingles vs Español

Ambos tienen sus pros y sus contras.

Español: Si alguien tiene deficiencias con el inglés puede llegar a tener problemas, más fácil poner nombres a las cosas. Traducir de los casos de uso que nos ha propuesto el usuario a código es mucho más natural. Hablar con el usuario o con otra gente del equipo se hace de una manera mucho más fluida al hablar el mismo idioma.

Lo que me mueve más a usar el inglés en muchos casos es que evito caracteres raros en nombres de variables y métodos llegando a los casos de ñ por ny y evitando tildes que pueden llegar a ser confusas, pueden surgir muchos problemas en cosas que no controlamos. Servidores, sistemas externos que tiene que ejecutar

Muchos problemas, soluciones y desarrollos de compañeros van en inglés, añadir el español al final es añadir una capa de complejidad que para mi no merece la pena.

De todas maneras cada proyecto tiene sus propias características y es algo que hay que hablar con el equipo.

Casos de Uso

Básicos

Gestión de los socios de la biblioteca

- Agregar nuevo socio
- Listar socios

Gestión del Catálogo

- Agregar Elemento al Catálogo
- Listado de Elementos
- Detalle de Elemento
- Edición de Elemento
- Borrado Elemento

Gestión de las Copias

- Agregar una copia a un elemento
- Ver las copias relacionadas con un elemento
- Borrar copia

Gestión de Préstamos

- Generar un préstamo seleccionado una copia y un usuario
- Devolución del préstamos

Para explorar nuevas funcionalidades

- Renovación de Préstamo (poco complejo)
- Reservar una Copia prestada (afecta a la renovación)
- Aviso de Expiración de Préstamo (Tarea Scheduled)
- Aumentar tipos de miembros. Infantil, Adulto, Infantil VIP, Adulto VIP (más préstamos y durante más tiempo)
- Tipo de Item (novedad, reciente adquisición: condiciones diferentes a la hora de préstamo)
- Aumentar Complejidad a la hora de Generar un préstamos. (Tipo de usuario, Tipo de Item, ¿reservado?)
- Validación de Datos antes de guardar o actualizar. (@Valid)

Definiciones Base de Datos

Items

- id
- type
- title
- author
- duration (la métrica de este campo depende del type)
- minimum_age
- released_at

Copies

- id
- item_id
- acquired_at
- reserved_by

Members

- id
- username
- email

Loans

user_id/copy_id no puede ser el id porque se puede prestar varias veces una copia a un usuario y añadir una fecha como id no me gusta.

- id
- user_id
- copy_id
- started_at
- expired_at
- returned_at