

“My Favourite Books”

Tu reto es crear una mini-aplicación web de tienda de libros utilizando un framework frontend como Vue.js, React o Angular (preferentemente Vue). Hemos considerado algunas funcionalidades básicas de todas las que pudieran estar para fines prácticos.

Para lograr esto, la mini-aplicación, se dividirá en 4 partes, y, deberás utilizar un servicio que hemos habilitado temporalmente para completar este reto.

API: bookstore.mgi.pe/api

Para más información del uso del api: bookstore.mgi.pe/docs

Ten en cuenta que es posible que para hacer uso de algunos servicios requieras crearte una cuenta y te autentiques, te recomendamos revisar la documentación.

Las interfaces son a criterio personal por lo cual podrás dar rienda suelta a tus ideas. Sin embargo, considerar, que debe existir un navbar donde se pueda acceder al catálogo de libros, colecciones personales, mis libros favoritos y registro de libros. Incluso, podrías considerar el uso de alguna plantilla de tu gusto o agregar funcionalidades que consideres sean necesarias o que no se hayan descrito en el reto.

Parte 1: Listado catálogo y registro de favoritos.

Deberás implementar un listado de todos los libros disponibles donde se debe visualizar los siguientes datos:

- Portada
- Título
- Autor
- Categoría
- Precio

Cada ítem mostrado deberá incluir un botón para agregar el libro a una colección personal, por tanto, al agregar un libro es posible elegir a que colección se agregará.

Parte 2: Registro de nuevos libros.

Deberás implementar un formulario mediante la cual el usuario pueda agregar un libro nuevo, la cual deberá tener los siguientes datos:

- Portada
- Título
- Autor
- Categoría
- Precio

Considerar, que la portada no es una imagen, sino más bien una URL desde donde se visualizará. Así mismo, considera que el registro de libros es abierto por tanto no requiere seguridad alguna. (Esto con fines prácticos claro está)

Parte 3: Personalización de componentes.

Crea un componente "PriceComponent" que reemplace al input de precio del formulario anterior que cumpla con las siguientes especificaciones:

1. Solo se debe permitir el ingreso de números
2. Se acepta 2 decimales como máximo, si no tuviera se completa con 0. (.00)
3. Para los números de miles se usará la coma
4. Para los números grandes se deberá usar el apostrofe, es decir, para los números con más de 6 dígitos en adelante

Ejemplo:

1'950,345.66

2'100'950,345.66

Entiéndase, que el formato debe aplicarse conforme se va ingresando los números, por tanto, no debe permitirse otros caracteres y debe ser lo más compatible posible.

Parte 4: Visualización de favoritos

Al igual que en el listado de libros, deberás mostrar un listado de las colecciones que usuario ha añadido, por tanto, deberás poder identificar a que usuario pertenece cada colección de libros.

Persistencia de Datos:

Utiliza alguna forma de almacenamiento de datos para mantener los datos de forma que persistan incluso después de recargar la página, como por ejemplo localStorage o el administrador de estados del framework (por ejemplo: pinia o vuex en vue), esto con el fin de evitar la llamada constante a algún servicio, esto es a criterio.

Diseño Responsivo:

Asegúrate de que tu solución sea completamente responsiva y se vea bien en dispositivos de diferentes tamaños, como teléfonos móviles y tabletas.

Accesibilidad:

Asegúrate de que la aplicación sea accesible para personas con discapacidades.

Optimización de Rendimiento:

Optimiza el rendimiento de la aplicación.

Consideraciones adicionales:

Utiliza Git para controlar la versión del código y realizar commits necesarios.

Implementar al menos dos pruebas unitarias en las funciones principales para garantizar el correcto funcionamiento.