**Requerimientos del Front-End**

El front-end debe ser intuitivo, responsivo y optimizado para garantizar una experiencia de usuario fluida para bibliotecarios, administradores y usuarios generales.

1. **Requisitos Funcionales**

*Gestión de Libros*

* **Listar libros**:  
  Mostrar un listado paginado de libros con información como título, autor, categoría, disponibilidad y ubicación.
* **Buscar libros**:  
  Barra de búsqueda para localizar libros por título, autor o ID.
* **Filtrar libros**:  
  Filtros dinámicos por categoría, estado (disponible/prestado) y rango de publicación.
* **Registrar nuevos libros**:  
  Formulario para registrar un nuevo libro, con campos como título, autor, ISBN, categoría y cantidad en inventario.
* **Editar detalles de libros**:  
  Opción de actualizar información del libro, como título o autor.
* **Eliminar libros**:  
  Posibilidad de eliminar registros obsoletos con confirmación previa.

*Gestión de Usuarios*

* **Listar usuarios**:  
  Mostrar una lista paginada con los detalles básicos de los usuarios (ID, nombre, correo, tipo de usuario).
* **Registrar usuarios**:  
  Formulario para registrar nuevos usuarios (estudiantes, profesores, etc.).
* **Editar información de usuarios**:  
  Posibilidad de modificar información del usuario, como nombre o tipo de usuario.
* **Suspender usuarios**:  
  Funcionalidad para suspender usuarios con justificación.

*Gestión de Préstamos y Devoluciones*

* **Registrar préstamos**:  
  Formulario para registrar un préstamo con validación automática de disponibilidad del libro y estado del usuario.
* **Registrar devoluciones**:  
  Función para registrar la devolución de un libro y calcular multas en caso de retraso.
* **Visualizar préstamos activos**:  
  Listado de préstamos activos con detalles como fecha de préstamo, fecha de devolución y estado del libro.
* **Historial de préstamos**:  
  Visualización del historial de préstamos por usuario o libro.

*Reportes y Estadísticas (opcional)*

* **Generar reportes**:  
  Modulo para crear reportes en formatos como PDF o Excel:
  + Inventario actual.
  + Préstamos activos.
  + Historial de un usuario.
* **Estadísticas visuales**:  
  Gráficos y tablas que muestren datos clave, como libros más prestados y usuarios activos.

1. **Requisitos No Funcionales**

*Interfaz de Usuario (UI)*

* **Diseño responsivo**:  
  Compatible con dispositivos móviles, tabletas y computadoras.
* **Estilo visual**:  
  Basado en un framework como Material Design o Bootstrap para asegurar uniformidad en colores, tipografías y botones.
* **Accesibilidad**:  
  Cumplir con estándares de accesibilidad (WCAG 2.1).

*Experiencia de Usuario (UX)*

* **Intuitividad**:  
  Navegación clara y sencilla entre módulos.
* **Feedback en tiempo real**:  
  Notificaciones para confirmaciones, errores y alertas mediante *toasts* o *modals*.

*Seguridad*

* Validación de entradas del usuario para evitar ataques XSS y SQL Injection.
* Sesiones protegidas mediante autenticación JWT.

*Rendimiento*

* Optimización del tiempo de carga utilizando técnicas como lazy loading y minimización de recursos.
* Manejo eficiente de listas paginadas para evitar sobrecarga en la interfaz.

*Integraciones*

* Consumo de APIs REST para consultas rápidas de libros y usuarios.
* Interacción con APIs SOAP para operaciones transaccionales complejas como préstamos y devoluciones.

1. **Herramientas y Tecnologías**

*Frameworks y Librerías (ejemplo)*

* **React.js**: Para construir una interfaz modular, eficiente y reactiva.
* **PrimeReact**: Para componentes visuales avanzados como tablas, gráficos y formularios.

*Gestión del Estado*

* **Redux**: Manejo de estado global para sincronizar datos entre diferentes vistas.

*Estilización*

* **CSS-in-JS (Styled Components)**: Para encapsular estilos.
* **Material-UI o Bootstrap**: Frameworks para asegurar un diseño profesional.

*Conexión con APIs (ejemplo)*

* **Axios**: Para consumir APIs REST y SOAP con manejo de errores y soporte para peticiones asíncronas.

*Ruteo (ejemplo)*

* **React Router**: Para navegación entre las diferentes páginas de la aplicación.

*Autenticación (ejemplo)*

* Implementación de Login y Logout con JWT, incluyendo roles de usuario para acceso controlado a módulos específicos.

1. **Estructura de Módulos**

*Bibliotecario/Admin*

1. Gestión de libros.
2. Gestión de usuarios.
3. Registro de préstamos y devoluciones.
4. Generación de reportes.

*Usuario Final (Estudiante/Profesor)*

1. Consulta de libros.
2. Visualización de historial de préstamos.
3. Reserva y renovación de préstamos.

*Dashboard de Administración*

1. Visualización de estadísticas clave.
2. Acceso a reportes.

**Prototipo Propuesto**

* **Home Page**: Con accesos directos a los módulos principales (libros, usuarios, préstamos).
* **Menú de Navegación**: Navbar fijo con opciones para cada módulo según el rol del usuario.
* **Vistas Específicas**:
  + Listado de datos en tablas con opciones de acción rápida.
  + Formularios con validación en tiempo real.