

**Sistema Integrado de Gestión de Biblioteca (SIGB) para la DGAC**

**Versión:** 1.0 (Prototipo Funcional)

**Fecha**: 10 de octubre de 2025

**Profesor a cargo:** Fabian Guajardo

**Autores:** Cristian Martínez, Renato Toro, Rodolfo Morales, Miguel Donoso

**Índice**

[**1. Resumen Ejecutivo 3**](#_heading=h.7nz60wabrbuv)

[1.1 Antecedentes y Contexto Institucional 4](#_heading=h.717wmtw2nn09)

[1.1.1 Demoras en la atención: 4](#_heading=h.rnbig546fvxn)

[1.1.2 Posibles errores humanos: 5](#_heading=h.gzj5wzt7s5v2)

[1.1.3 Dificultad para obtener reportes fiables: 5](#_heading=h.c9b0uieme5es)

[1.1.4 Falta de trazabilidad y control: 5](#_heading=h.ve8594vrc594)

[1.2 Objetivos del Proyecto 5](#_heading=h.yoanyta1yuh0)

[1.2.1 Objetivo General: 5](#_heading=h.qhiggl4o9cg6)

[1.2.2 Objetivos específicos: 5](#_heading=h.5of87jwby8fr)

[1.3 Alcance Detallado del Sistema 6](#_heading=h.d0ys0inkfoig)

[**2. Objetivos Medibles de la Arquitectura (KPIs) 7**](#_heading=h.pd6q4uax8c3e)

[**3. Módulos Funcionales 8**](#_heading=h.3eywedzc5vdy)

[3.1 Circulación 8](#_heading=h.ub6efnv748w3)

[3.2 Catalogación 9](#_heading=h.i9q3bqvtz9ye)

[3.3 Catálogo Público (OPAC) 10](#_heading=h.azcn0cuxagkq)

[3.4 Administración del Sistema 11](#_heading=h.r8ihpgfcjden)

[**4. Modelo de Datos (Esquema Lógico) 13**](#_heading=h.1gp4gvxl4pxw)

[4.1 Principales entidades: 13](#_heading=h.5cx1iuefa9xn)

[4.2 Políticas: 14](#_heading=h.wvtmw4cm57pv)

[4.2.1 Integridad referencial: 14](#_heading=h.rfdrwgt7we3)

[4.2.2 Eliminación lógica: 14](#_heading=h.1hcixga0hxs2)

[4.2.3 Índices por título, autor y materia: 14](#_heading=h.9dyd8nfp2560)

[4.2.4 Control de duplicidad: 14](#_heading=h.vq2iytd4brje)

[4.2.5 Política de auditoría: 14](#_heading=h.nwg89sbhures)

[4.2.6 Manejo de transacciones atómicas: 14](#_heading=h.gcbbmhsrtpgp)

[**5. Reglas de Negocio Clave 15**](#_heading=h.10dsmy7mc23s)

[5.1 Préstamos únicos por ejemplar: 15](#_heading=h.m0my7l3dlsy)

[5.2 Límites de préstamos según rol de usuario: 15](#_heading=h.x49xw1d9601o)

[5.3 Validación de disponibilidad: 15](#_heading=h.iyw1jliw2axb)

[5.4 Registro automático de transacciones: 15](#_heading=h.ptsscjfge5at)

[5.5 Restricciones sobre eliminación: 16](#_heading=h.p5txcrqys90q)

[5.6 Reservas y prioridad: 16](#_heading=h.useibx3g41nt)

[5.7 Penalizaciones automáticas (futuro): 16](#_heading=h.c0r9bxv97mc8)

[**6. Decisiones Arquitectónicas Clave (ADRs) 16**](#_heading=h.oga2rokswflp)

[**7. Requisitos No Funcionales (NFRs) 17**](#_heading=h.uw1u71b3x8d7)

[**8. Restricciones y Supuestos 17**](#_heading=h.8oc6b3jxlytv)

[**9. Plan de Pruebas (Prototipo) 18**](#_heading=h.yd5w46j6l02w)

[10. Herramientas y Tecnologías Utilizadas 19](#_heading=h.nekziw465gpe)

[**11. Próximos Pasos y Escalabilidad 20**](#_heading=h.7qdc49b6j2fd)

# 1. Resumen Ejecutivo

Este documento detalla la arquitectura del Sistema Integrado de Gestión Bibliotecaria (SIGB), un prototipo funcional desarrollado como parte del proyecto Capstone de Duoc UC para la DGAC.

El objetivo principal del sistema es digitalizar y automatizar procesos manuales asociados a la gestión bibliotecaria, optimizando la eficiencia operativa y mejorando la experiencia de los usuarios internos y externos.

Componentes principales de la arquitectura

Frontend: Aplicación Web SPA (Single Page Application), diseñada con una interfaz fluida, responsive y accesible, orientada a brindar una experiencia de usuario clara y eficiente en distintos dispositivos.

Backend: API RESTful desarrollada en Django, que centraliza la lógica de negocio, gestiona autenticación, autorización, roles de usuario y operaciones CRUD para los distintos módulos.

Base de Datos: SQLite en ambiente de desarrollo local para simplicidad y portabilidad, con planificación de migración a PostgreSQL para producción, asegurando robustez, escalabilidad y soporte transaccional.

Integraciones y despliegue futuro

Se contempla integración futura con SSO corporativo (OIDC/SAML), notificaciones vía SMTP y conexión con Data Warehouse (DWH) para análisis y reportes avanzados.

El sistema está diseñado para ser contenedorizada con Docker, facilitando su despliegue en entornos Kubernetes para lograr portabilidad, escalabilidad horizontal y alta disponibilidad.

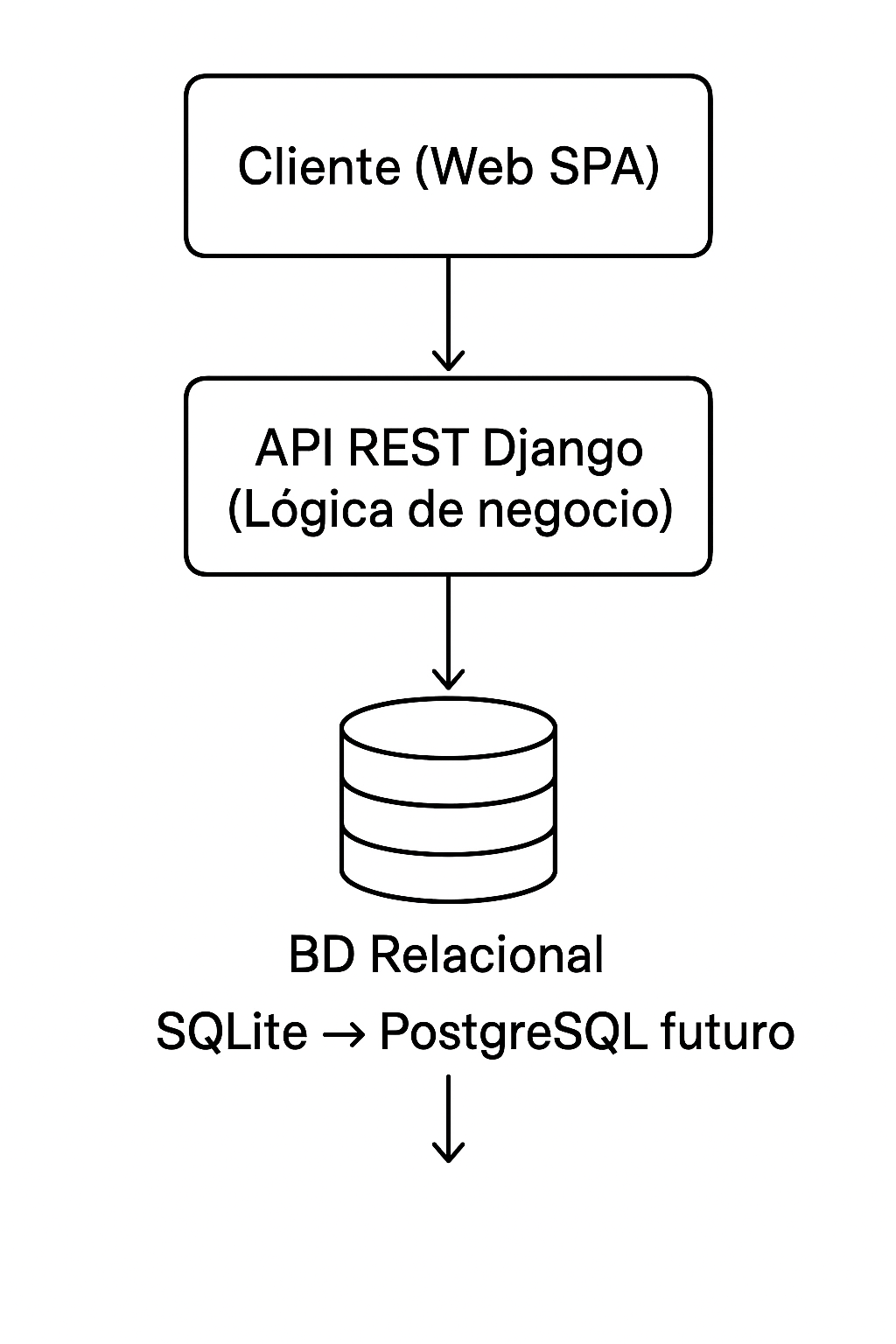
La arquitectura modular permite extender funcionalidades sin afectar la estabilidad del sistema base.

Consideraciones técnicas clave

Separación clara entre presentación, lógica de negocio y persistencia de datos.

Diseño orientado a servicios y escalabilidad progresiva.

Cumplimiento con buenas prácticas de seguridad y gestión de usuarios institucionales.



## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 1.1 Antecedentes y Contexto Institucional

Actualmente, la biblioteca de la DGAC gestiona préstamos, devoluciones, inventario y usuarios de manera totalmente manual, lo que genera varias dificultades operativas y afecta la experiencia de los usuarios:

### 1.1.1 Demoras en la atención:

Los procesos manuales de registro y búsqueda de materiales generan esperas prolongadas para los usuarios, especialmente en horarios de alta afluencia.

### 1.1.2 Posibles errores humanos:

La entrada manual de datos aumenta el riesgo de equivocaciones en los registros de préstamos, devoluciones o inventario, lo que puede derivar en inconsistencias y pérdida de información.

### 1.1.3 Dificultad para obtener reportes fiables:

La generación de estadísticas sobre préstamos, disponibilidad de materiales o historial de usuarios requiere procesos adicionales de recopilación y verificación, lo que limita la capacidad de análisis y toma de decisiones por parte del personal.

### 1.1.4 Falta de trazabilidad y control:

Al no existir registros automatizados, no se puede auditar fácilmente quién realiza cada acción ni cuándo, afectando la transparencia y la seguridad de la información.

El SIGB busca un sistema centralizado y automatizado, que permita trazabilidad y control de los procesos, respetando la privacidad de datos reales.

## 1.2 Objetivos del Proyecto

### 1.2.1 Objetivo General:

Digitalizar y automatizar los procesos de gestión bibliotecaria en la DGAC, optimizando eficiencia, seguridad y trazabilidad de todas las operaciones.

### 1.2.2 Objetivos específicos:

Digitalizar procesos clave

Automatizar préstamos, devoluciones, catalogación y consultas de materiales bibliográficos.

Garantizar que cada operación quede registrada en el sistema, reduciendo errores humanos y mejorando la confiabilidad de los datos.

Desarrollar una interfaz web accesible y responsiva

Proporcionar un diseño intuitivo y fácil de usar para usuarios de distintos perfiles y niveles de habilidad tecnológica.

Asegurar compatibilidad con distintos dispositivos (computadoras, tablets y móviles) para facilitar el acceso remoto.

Implementar roles claros con permisos diferenciados

Definir niveles de acceso (Administrador, Cargador, Lector) con funcionalidades específicas.

Garantizar seguridad y control, evitando operaciones no autorizadas sobre los datos del sistema.

Garantizar integridad y trazabilidad de la información

Registrar todas las transacciones, modificaciones y accesos en tablas de auditoría.

Mantener integridad referencial y eliminación lógica de registros para asegurar consistencia histórica.

Desarrollar un sistema modular, escalable y seguro

Permitir la incorporación futura de nuevas funcionalidades sin afectar al núcleo del sistema.

Escalar la infraestructura para soportar mayor número de usuarios o transacciones.

Implementar buenas prácticas de seguridad, incluyendo cifrado de datos y control de acceso basado en roles.

## 1.3 Alcance Detallado del Sistema

Incluye:

Circulación: Gestión completa de préstamos y devoluciones, incluyendo registro automático de fechas, control de plazos según el tipo de usuario, prevención de préstamos duplicados y notificaciones internas de vencimiento o retraso.

Catalogación: Registro, edición y eliminación de materiales como libros, revistas, mapas y documentos aeronáuticos. Soporta metadatos estandarizados (título, autor, editorial, materia, ISBN) y categorización por tipo de material para facilitar búsquedas y reportes.

Catálogo Público en Línea (OPAC): Consulta de materiales disponibles por filtros de título, autor, materia y tipo de material. Interfaz responsiva y amigable, compatible con dispositivos móviles, y planificación de búsqueda avanzada modularizada para integración futura con motores como Elasticsearch.

Administración de usuarios y reportes: Gestión de perfiles y roles (Administrador, Cargador, Lector), trazabilidad de acciones mediante auditoría con timestamp y hash, y generación de reportes sobre préstamos, devoluciones, reservas y actividad de usuarios.

Gestión de roles y autenticación segura: Sistema de permisos diferenciados según rol, garantizando que cada usuario acceda únicamente a las funciones permitidas.

No incluye (Futuro):

Integración con SSO corporativo, ERP institucional o Data Warehouse (DWH) reales.

Despliegue en infraestructura en la nube; el prototipo opera en entorno local.

Uso de datos reales de usuarios o materiales, para proteger la privacidad y cumplir con regulaciones internas.

# 2. Objetivos Medibles de la Arquitectura (KPIs)

| **Métrica** | **Valor Esperado** |
| --- | --- |
| Rendimiento OPAC Latencia | p 95 < 500 m |
| Eficiencia | Registro de préstamo < 30 s |
| Disponibilidad | 99.5% en horario hábil |
| Resiliencia | RPO ≤ 15 min / RTO ≤ 1 h |
| Seguridad | RBAC por roles, cifrado de datos |

# 3. Módulos Funcionales

## 3.1 Circulación

El módulo de Circulación gestiona todas las operaciones relacionadas con préstamos y devoluciones de manera automatizada, garantizando trazabilidad y control. Sus principales funciones incluyen:

Registro de préstamos:

Asociación automática de ejemplares a usuarios.

Control de disponibilidad para evitar préstamos duplicados.

Registro de fechas de inicio y vencimiento según tipo de usuario (lector, administrativo, externo).

Generación de notificaciones internas de vencimiento o retraso (planificado para futuras versiones).

Registro de devoluciones:

Actualización automática del estado del ejemplar.

Cálculo de multas o sanciones en caso de retrasos (futuro).

Actualización de historial de préstamos del usuario.

Configuración de plazos por rol de usuario:

Diferenciación de días máximos de préstamo según rol.

Posibilidad de configurar excepciones o extensiones por parte del administrador.

Prevención de préstamos duplicados:

Sistema que verifica disponibilidad y evita que un ejemplar sea prestado a más de un usuario simultáneamente.

| **Método** | **URL** | **Función** | **Estado** |
| --- | --- | --- | --- |
| GET | /api/prestamos/ | Listar préstamos | Implementado |
| POST | /api/prestamos/ | Crear préstamo | Implementado |
| PUT/PATCH | /api/prestamos/{id}/ | Actualizar estado | Implementado |
| DELETE | /api/prestamos/{id}/ | Cancelar préstamo | Futuro |

## 3.2 Catalogación

El módulo de Catalogación permite gestionar de manera estructurada todos los materiales disponibles en la biblioteca, asegurando consistencia y trazabilidad de los datos. Sus principales funciones incluyen:

Registro de materiales:

Creación de registros para libros, revistas, mapas, materiales aeronáuticos y otros documentos.

Inclusión de metadatos estandarizados (ISBN, autor, año, materia, editorial) para facilitar búsquedas y reportes.

Control de duplicados para mantener la integridad de la base de datos.

Edición y eliminación controlada:

Modificación de registros existentes con historial de cambios.

Eliminación lógica de registros para mantener trazabilidad histórica.

Acceso restringido según roles de usuario (administrador, cargador).

| **Método** | **URL** | **Función** | **Estado** |
| --- | --- | --- | --- |
| GET | /api/materiales/ | Listar materiales | Implementado |
| POST | /api/materiales/ | Crear material | Implementado |
| PUT/PATCH | /api/materiales/{id}/ | Editar material | Implementado |
| DELETE | /api/materiales/{id}/ | Eliminar material | Implementado |

## 3.3 Catálogo Público (OPAC)

El módulo de Catálogo Público (OPAC) permite que los usuarios consulten y exploren los materiales de la biblioteca de manera eficiente y amigable, ofreciendo información completa y filtrada según distintos criterios. Sus principales funciones incluyen:

Consultas filtradas:

Búsqueda por título, autor, material, tipo de material y año de publicación.

Filtros combinados para obtener resultados más precisos.

Soporte para paginación de resultados para mejorar rendimiento.

Visualización de disponibilidad de ejemplares:

Información en tiempo real sobre la cantidad de ejemplares disponibles.

Indicadores de préstamos activos y reservas pendientes.

Navegación responsiva:

Interfaz accesible desde dispositivos móviles, tablets y escritorio.

Diseño intuitivo para facilitar la exploración del catálogo.

| **Método** | **URL** | **Función** | **Estado** |
| --- | --- | --- | --- |
| GET | /api/opac/ | Consultar catálogo | Implementado |
| GET | /api/opac/?filtros | Filtrado avanzado | Futuro (Elasticsearch) |

## 3.4 Administración del Sistema

Este módulo permite gestionar usuarios, roles y permisos, así como mantener un registro completo de las acciones realizadas en el sistema, asegurando trazabilidad y cumplimiento de políticas internas. Sus principales funciones incluyen:

Gestión de usuarios:

Creación, edición y eliminación de usuarios.

Control de estados (activo, inactivo).

Asignación de roles según perfil (Administrador, Cargador, Lector).

Roles y permisos diferenciados:

Cada rol tiene acceso exclusivo a funciones específicas del sistema.

Prevención de acciones no autorizadas mediante validaciones a nivel de backend.

Auditoría y trazabilidad:

Registro automático de todas las acciones críticas (préstamos, devoluciones, modificaciones de datos).

Cada evento se almacena con timestamp, usuario responsable y hash de verificación.

Permite generar reportes históricos y garantizar la integridad de la información.

Ejemplo de flujo de auditoría:

Usuario realiza un préstamo → Evento registrado con timestamp y hash → Se almacena referencia en la tabla Auditoría → Posibilidad de generar reportes o exportar información para supervisión.

Endpoints REST (Resumen):

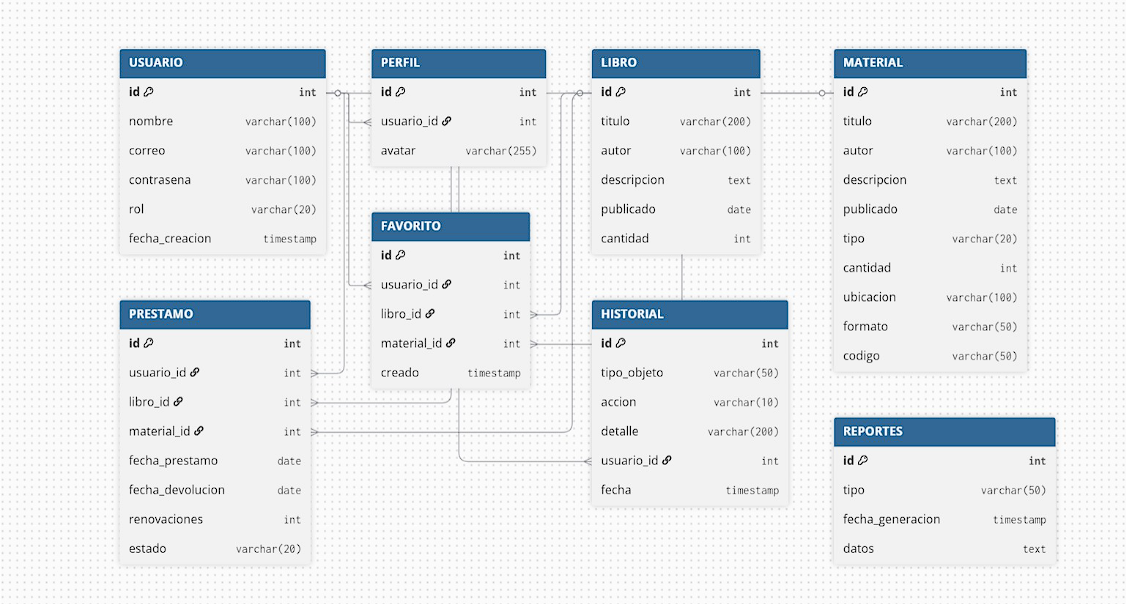
| **Método** | **URL** | **Función** | **Estado** |
| --- | --- | --- | --- |
| GET | /api/usuarios/ | Listar usuarios | Implementado |
| POST | /api/usuarios/ | Crear usuario | Implementado |
| PUT/PATCH | /api/usuarios/{id}/ | Modificar usuario | Implementado |
| DELETE | /api/usuarios/{id}/ | Eliminar usuario | Futuro |

# 

# 4. Modelo de Datos (Esquema Lógico)

## 4.1 Principales entidades:

| **Entidad** | **Atributos** |
| --- | --- |
| Usuario | id, nombre, correo, contraseña, rol, fecha\_creacion |
| Perfil | id, usuario\_id (FK), avatar |
| Libro | id, título, autor, descripción, publicado, cantidad |
| Material | id, título, autor, descripción, publicado, tipo, cantidad, ubicación, formato, código |
| Préstamo | id, usuario id (FK), libro id (FK, nullable), material id (FK, nullable), fecha préstamo, fecha devolución, renovaciones, estado |
| Favorito | id, tipo objeto, acción, detalle, usuario\_id (FK), fecha |
| Historial | id, tipo, fecha generación, datos |
| Reportes | id, tipo, fecha generación, datos |



## 

## 4.2 Políticas:

### 4.2.1 Integridad referencial:

Todas las relaciones entre entidades (usuarios, materiales, ejemplares, préstamos, reservas) están controladas mediante claves foráneas. Esto asegura la consistencia de los datos y evita registros huérfanos (por ejemplo, no se puede eliminar un usuario que tenga préstamos activos).

### 4.2.2 Eliminación lógica:

En lugar de eliminar registros de forma física, se utiliza un campo de estado (activo = True/False) para preservar el historial y la trazabilidad de la información, especialmente útil para auditorías futuras.

### 4.2.3 Índices por título, autor y materia:

Se crean índices en campos de búsqueda frecuentes para optimizar el rendimiento de consultas en el OPAC y los módulos internos de catalogación.

### 4.2.4 Control de duplicidad:

Se aplican restricciones únicas en campos clave (como ISBN y códigos de ejemplar) para evitar registros repetidos y asegurar la calidad de los datos.

### 4.2.5 Política de auditoría:

Cada transacción relevante (creación, modificación o eliminación lógica) genera un registro en la tabla de auditoría, con información sobre el usuario, fecha y tipo de acción.

### 4.2.6 Manejo de transacciones atómicas:

Las operaciones críticas (como crear un préstamo o modificar el estado de un ejemplar) se realizan de forma atómica para evitar estados inconsistentes ante fallas.

# 

# 5. Reglas de Negocio Clave

## 5.1 Préstamos únicos por ejemplar:

Cada ejemplar puede estar asociado únicamente a un préstamo activo en un momento dado. Esto evita conflictos en el inventario, garantiza que cada material tenga un seguimiento trazable y permite auditorías precisas sobre quién posee cada recurso.

## 5.2 Límites de préstamos según rol de usuario:

Para asegurar una distribución equilibrada de los recursos y priorizar a distintos tipos de usuarios, se aplican límites específicos:

* **Lector:** máximo 3 ejemplares por 7 días.
* **Editor:** máximo 5 ejemplares por 14 días.
* **Administrador:** sin límite estricto, pero con registro obligatorio de todas las transacciones.

Esta regulación permite controlar el stock disponible, evitando acaparamiento de materiales y optimizando la disponibilidad para todos los usuarios.

## 5.3 Validación de disponibilidad:

Antes de confirmar cualquier préstamo, el sistema verifica que el ejemplar no tenga reservas previas, no esté actualmente prestado y se encuentre en estado “disponible”. Esta medida reduce errores humanos, garantiza el cumplimiento de las reglas y mejora la experiencia del usuario.

## 5.4 Registro automático de transacciones:

Todas las acciones que modifican el estado de los recursos —préstamos, devoluciones, modificaciones de ejemplares o usuarios— se registran automáticamente en la tabla de auditoría. Cada registro incluye usuario responsable, timestamp y tipo de acción, asegurando trazabilidad completa y facilitando controles internos.

## 5.5 Restricciones sobre eliminación:

No se permite eliminar físicamente usuarios con préstamos activos ni ejemplares actualmente prestados. Todos los registros deben regularizar su estado antes de ser dados de baja, manteniendo la integridad operativa y evitando inconsistencias en el inventario.

## 5.6 Reservas y prioridad:

Cuando un ejemplar está reservado, solo el usuario que generó la reserva puede efectuar el préstamo una vez el material esté disponible. Esto asegura un flujo controlado y justo en la gestión de reservas, evitando conflictos y superposiciones.

## 5.7 Penalizaciones automáticas (futuro):

Se proyecta implementar un sistema automático de multas o bloqueos de cuentas para usuarios que no devuelvan los ejemplares dentro del plazo estipulado. Esta funcionalidad fomentará el cumplimiento de los tiempos de préstamo y la responsabilidad en el uso de los recursos.

# 6. Decisiones Arquitectónicas Clave (ADRs)

# 

| **ID** | **Descripción** |
| --- | --- |
| ADR-001 | PostgreSQL en producción por transacciones y escalabilidad. |
| ADR-002 | API REST con DRF para compatibilidad y caché HTTP. |
| ADR-003 | Contenedores Docker para portabilidad y despliegue. |
| ADR-004 | Búsqueda avanzada modularizada para integración futura con Elasticsearch. |

# 7. Requisitos No Funcionales (NFRs)

| **Categoría** | **Descripción** | **Métrica** |
| --- | --- | --- |
| Rendimiento | Respuesta de búsqueda | p95 < 500 ms |
| Escalabilidad | Multiproceso en contenedores | Autoscaling horizontal |
| Disponibilidad | Servicio activo | 99.5% |
| Seguridad | Autenticación, cifrado, auditoría | RBAC + TLS |
| Mantenibilidad | Modularidad y documentación | 90% funciones críticas |

# 8. Restricciones y Supuestos

Datos ficticios: Se utilizan datos simulados para proteger la privacidad institucional y cumplir con normativas internas de seguridad.

Entorno local: El desarrollo y pruebas se realizan en un entorno local (localhost:8000) utilizando Django como framework principal.

Limitaciones de infraestructura: El proyecto no depende de servidores en la nube en esta etapa, lo que restringe su acceso a entornos externos pero facilita la gestión técnica inicial.

# 

# 9. Plan de Pruebas (Prototipo)

| **Tipo** | **Objetivo** | **Resultado esperado** |
| --- | --- | --- |
| Funcional | CRUD completo | 100% OK |
| Desempeño | Búsquedas simultáneas | < 1s promedio |
| Seguridad | Validación de roles | Sin vulneraciones |
| Usabilidad | Flujo intuitivo Likert | ≥4, aprobado |

### 

# 10. Herramientas y Tecnologías Utilizadas

| Componente | Herramienta / Tecnología | Justificación |
| --- | --- | --- |
| Lenguaje de programación | Python 3.11+ | Fácil integración con Django, amplio soporte de librerías y comunidad activa |
| Framework Backend | Django + Django REST Framework | Manejo de autenticación, roles y APIs RESTful “listo para usar” |
| Base de Datos | SQLite (desarrollo) $\rightarrow$ PostgreSQL (producción) | SQLite simplifica el prototipo local; PostgreSQL permite escalabilidad, concurrencia y transacciones robustas |
| Frontend | HTML, CSS, JavaScript (SPA) | Interfaz intuitiva y responsiva; compatible con distintos dispositivos |
| Contenerización | Docker | Portabilidad, despliegue reproducible y fácil escalabilidad |
| Control de versiones | Git / GitHub | Gestión de código y colaboración en equipo |
| Otros | Visual Studio Code, Postman, Figma | Desarrollo, pruebas de API y diseño de interfaz |

# 

# 11. Próximos Pasos y Escalabilidad

Parametrización dinámica de préstamos:

Se implementará un sistema flexible que permita configurar los límites de préstamos, plazos y penalizaciones según el rol del usuario, facilitando la adaptación a distintas políticas internas de la biblioteca.

Sistema de reportes con gráficos interactivos:

Se desarrollarán reportes visuales para monitorear estadísticas clave, como préstamos activos, materiales más solicitados, usuarios más frecuentes y tasas de devolución. Esto permitirá tomar decisiones basadas en datos y mejorar la gestión operativa.

Documentación de APIs y modelo de datos:

Se entregará documentación completa y actualizada de las APIs REST y del modelo de datos, facilitando la integración con otros sistemas, soporte para nuevos desarrollos y mantenimiento a largo plazo.