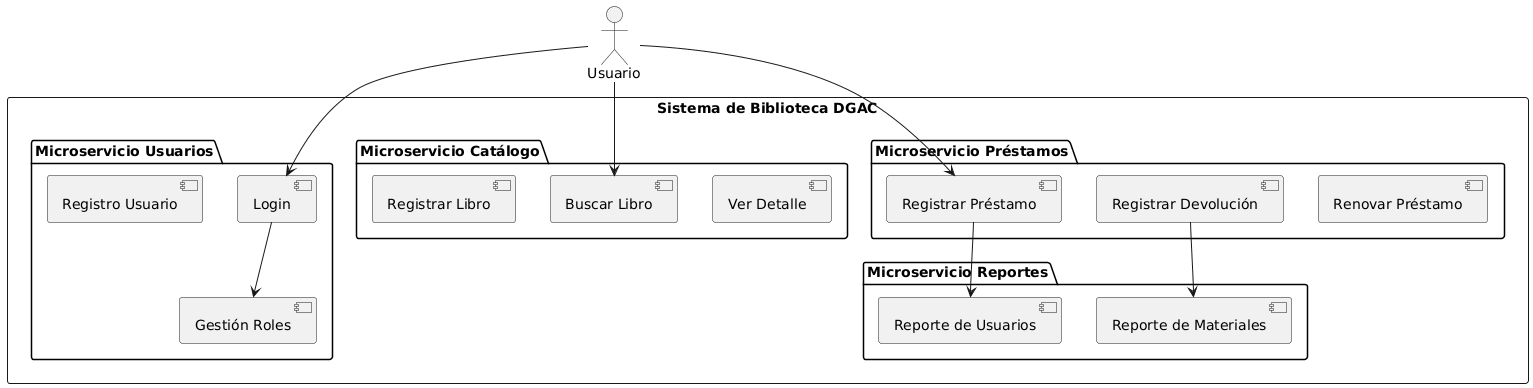
Arquitectura de Software

### **3.1 Introducción**

La arquitectura seleccionada para el desarrollo del *Sistema de Biblioteca Dirección General de Aeronáutica Civil(DGAC)* corresponde a una **arquitectura de microservicios**. Esta decisión se fundamenta en la necesidad de contar con un sistema modular, escalable y fácil de mantener, en el que cada funcionalidad crítica (usuarios, catálogo, préstamos y reportes) pueda desarrollarse, desplegarse y mantenerse de manera independiente.

### **3.2 Diagrama de Arquitectura**

A continuación se presenta el diagrama de componentes UML que refleja la distribución de microservicios y la interacción con el usuario final:

**

Los microservicios definidos son:

* **Usuarios**: Gestión de registro, login, roles y perfiles.
* **Catálogo**: Administración de libros y materiales bibliográficos.
* **Préstamos**: Control de préstamos, devoluciones y renovaciones.
* **Reportes**: Generación de reportes estadísticos de uso y circulación.

### **3.3 Definición de APIs REST**

**Microservicio Usuarios**

* POST /usuarios → Registrar usuario
* POST /login → Iniciar sesión
* GET /usuarios/{id} → Consultar usuario
* PUT /usuarios/{id} → Actualizar usuario
* DELETE /usuarios/{id} → Eliminar usuario

**Microservicio Catálogo**

* POST /libros → Registrar libro
* GET /libros → Listar libros
* GET /libros/{id} → Ver detalle
* PUT /libros/{id} → Modificar libro
* DELETE /libros/{id} → Eliminar libro

**Microservicio Préstamos**

* POST /prestamos → Registrar préstamo
* PUT /prestamos/{id}/devolucion → Registrar devolución
* PUT /prestamos/{id}/renovacion → Renovar préstamo
* GET /prestamos → Listar préstamos activos

**Microservicio Reportes**

* GET /reportes/usuarios → Reporte de usuarios activos
* GET /reportes/materiales → Reporte de circulación de materiales

### **3.4 Modelo de Datos**

Cada microservicio cuenta con su propio modelo de datos en **MySQL**, con el fin de mantener independencia y escalabilidad.

**Usuarios**

* Usuario (*id\_usuario*, nombre, email, contraseña, rol, fecha\_creacion)

**Catálogo**

* Libro (*id\_libro*, título, autor, año, categoría, estado)

**Préstamos**

* Prestamo (*id\_prestamo*, usuario\_id, libro\_id, fecha\_prestamo, fecha\_devolucion, estado)

**Reportes**

* Se generan a partir de las tablas anteriores, sin necesidad de entidades adicionales.

*(Aquí se recomienda incluir un diagrama ER sencillo con estas entidades y relaciones).*

### **3.5 Tecnologías y Herramientas**

* **Lenguaje y Framework Backend:** Python – Django Rest Framework
* **Base de datos:** MySQL
* **Comunicación:** APIs REST (JSON sobre HTTP)
* **Control de versiones:** Git/GitHub
* **Entorno de despliegue inicial:** Localhost (MVP), con proyección a contenedores Docker

### **3.6 Seguridad**

* Autenticación mediante **tokens JWT**.
* Contraseñas encriptadas en la base de datos.
* Gestión de roles (Administrador / Usuario lector).
* Restricción de acceso a endpoints según permisos.

### **3.7 Escalabilidad futura**

El diseño modular de microservicios permite:

* Incorporar nuevas funcionalidades sin afectar los módulos existentes.
* Migrar a entornos de despliegue con **contenedores (Docker)**.
* Integrar un **API Gateway** para centralizar accesos y balancear carga.
* Extender la solución a otras bibliotecas de la DGAC u otras instituciones.