**Módulo de Inicio de Sesión**

**Módulo de Autenticación:**

* **Funciones:**
* Validación de credenciales (usuario y contraseña).
* Generación de tokens JWT (JSON Web Tokens) para autenticación posterior.
* Almacenamiento seguro de hashes de contraseñas (utilizando algoritmos como bcrypt).
* Implementación de autenticación de dos factores (por ejemplo, usando TOTP o email).
* Gestión de tokens de refresco para mantener las sesiones activas.
* **Tecnologías:**
* Framework de autenticación como Flask-Login o Django-OAuth-Toolkit.
* Base de datos para almacenar usuarios y sus credenciales.

**Módulo de Autorización:**

* **Funciones:**
* Definición de roles y permisos para cada perfil de usuario.
* Verificación de permisos para cada solicitud del usuario.
* Gestión de matrices de acceso para controlar las acciones permitidas.
* **Tecnologías:**
* Framework de autorización como Casbin o OAuth2-Server.

**Módulo de Usuarios:**

* **Funciones:**
* Gestión del ciclo de vida de los usuarios (registro, actualización, eliminación).
* Asociación de usuarios con perfiles y roles.
* Almacenamiento de información de perfil del usuario (nombre, correo electrónico, etc.).
* **Tecnologías:**
* ORM (Object Relational Mapper) para interactuar con la base de datos.

**Módulo de Integración:**

* **Funciones:**
* Interacción con sistemas externos (control de acceso físico, otros módulos del sistema).
* Adaptación a diferentes protocolos de comunicación (REST, MQTT, etc.).
* **Tecnologías:**
* Librerías específicas para cada sistema externo.

### **Estructura de la Base de Datos (4FN)**

**Tabla Usuarios:**

* id\_usuario (PK)
* nombre
* correo\_electronico
* hash\_contraseña
* perfil (dueño, veterinario, trabajador)

**Tabla Permisos:**

* id\_permiso (PK)
* nombre\_permiso
* descripcion

**Tabla Roles:**

* id\_rol (PK)
* nombre\_rol

**Tabla Usuarios\_Roles:**

* id\_usuario (FK)
* id\_rol (FK)

**Tabla Permisos\_Roles:**

* id\_rol (FK)
* id\_permiso (FK)