**1. Autenticación y Autorización**

**Implementación de Autenticación**

La autenticación se implementa utilizando JWT (JSON Web Tokens) y bcrypt para el hash de contraseñas. El sistema incluye:

1. Registro de Usuarios (POST /auth/register)

* Parámetros:

      {

        "fullName": "string",

        "email": "string",

        "password": "string",

        "dni": "string",

        "phone": "string",

        "roles": ["client" | "admin" | "technician"]

      }

* Respuesta:

{

"id": "uuid",

"fullName": "string",

"email": "string",

"dni": " string ",

"phone": " string ",

"isActive": boolean,

"roles": [ “string”, “string” ],

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6ImI2NDc3NzRiLTBmMzYtNDgzYi04MmU5LTY1MDg0OGY2OWU5MCIsInJvbGVzIjpbImNsaWVudCIsImFkbWluIiwidGVjaG5pY2lhbiJdLCJpYXQiOjE3NDcwMjY5MTAsImV4cCI6MTc0NzAzMDUxMH0.YTWdpPeci8U4VvlQyrGUkoqR9wD8BQvJ3NslA31xn9I"

}

* Usuario creado con sus datos
* Token JWT para autenticación

1. Inicio de Sesión (POST /auth/login)

* Parámetros:

      {

        "email": "string",

        "password": "string"

      }

* Respuesta:
  + Datos del usuario
  + Token JWT

1. **Verificación de Estado de Autenticación** (GET /auth/check-auth-status)

* Requiere token JWT en el header
* Verifica si el token es válido y el usuario está activo

1. **Dashboard de Administrador** (GET /auth/admin-dashboard)

* Requiere token JWT y rol de administrador
* Proporciona acceso a métricas y funcionalidades administrativas

1. **Dashboard de Ventas** (GET /auth/sales-dashboard)

* Requiere token JWT y rol apropiado
* Muestra métricas y datos de ventas

**Sistema de Roles**

La API implementa un sistema de roles con tres niveles:

* client: Usuarios que pueden alquilar equipos
* admin: Administradores con acceso total
* technician: Técnicos que manejan entregas y reparaciones
* technicianlab: Técnicos del laboratorio que se encargan de revisar a detalles los equipos y darles mantenimiento
* salesperson: Son los que arriendan sus equipos a los clientes

**2. Endpoints por Módulo**

**Módulo de Usuarios (/users)**

1. **Obtener todos los usuarios** (GET /users)

* Requiere autenticación
* Retorna lista de usuarios

1. **Obtener usuario por ID** (GET /users/:id)

* Requiere autenticación
* Retorna datos del usuario específico

1. **Obtener usuario por email** (GET /users/email/:email)

* Requiere autenticación
* Retorna datos del usuario por email

1. **Modificar usuario** (PATCH /users/:id)

* Requiere autenticación
* Parámetros:

     {

        "fullName": "string",

        "email": "string",

        "password": "string"

    }

1. **Eliminar usuario** (DELETE /users/:id)

* Requiere autenticación y permisos de administrador

**Módulo de Roles (/roles)**

1. **Obtener todos los roles** (GET /roles)

* Requiere autenticación

1. **Crear rol** (POST /roles)

* Requiere autenticación y permisos de administrador
* Parámetros:

     {

        "name": "string",

        "description": "string",

        "permissions": ["string"]

      }

1. **Obtener rol por ID** (GET /roles/:id)

* Requiere autenticación

1. **Modificar rol** (PATCH /roles/:id)

* Requiere autenticación y permisos de administrador
* Parámetros iguales a los de creación

1. **Eliminar rol** (DELETE /roles/:id)

* Requiere autenticación y permisos de administrador

**Módulo de Equipos (/equipments)**

1. **Crear equipo** (POST /equipments)

* Requiere autenticación y permisos adecuados
* Parámetros:

     {

        "name": "string",

        "type": "computer" | "printer" | "phone",

        "brand": "string",

        "model": "string",

        "description": "string",

        "price": number,

        "stock": number,

        "warrantyPeriod": "string",

        "releaseDate": "string",

        "computerSpecs" | "printerSpecs" | "phoneSpecs": {

*// Especificaciones según el tipo*

         }

      }

1. **Obtener todos los equipos** (GET /equipments)

* Requiere autenticación

1. **Obtener equipo por ID** (GET /equipments/:id)

* Requiere autenticación

1. **Modificar equipo** (PATCH /equipments/:id)

* Requiere autenticación y permisos
* Parámetros similares a la creación

1. **Eliminar equipo** (DELETE /equipments/:id)

* Requiere autenticación y permisos adecuados

**Módulo de Alquileres (/rentals)**

1. **Solicitar alquiler** (POST /rentals/request)

* Requiere autenticación
* Parámetros:

json

Apply

     {

       "equipmentId": "string",

       "startDate": "string",

       "endDate": "string"

     }

1. **Obtener contratos de alquiler** (GET /rentals/my-contracts)

* Requiere autenticación
* Retorna lista de contratos del usuario

1. **Obtener métricas** (GET /rentals/metrics)

* Requiere autenticación y permisos adecuados
* Retorna métricas de alquileres

**Módulo de Entregas (/deliveries)**

1. **Crear entrega** (POST /deliveries)

* Requiere autenticación y rol de técnico
* Parámetros:

     {

        "rental": { "id": "string" },

        "technician": { "id": "string" },

        "client": { "id": "string" },

        "actDocumentUrl": "string",

        "clientSignatureUrl": "string",

        "visualObservations": "string",

        "technicalObservations": "string"

      }

1. **Obtener todas las entregas** (GET /deliveries)

* Requiere autenticación

1. **Obtener entrega por ID** (GET /deliveries/:id)

* Requiere autenticación

1. **Modificar entrega** (PATCH /deliveries/:id)

* Requiere autenticación y permisos
* Parámetros:

     {

        "status": "string",

        "technicalObservations": "string"

  }

1. **Eliminar entrega** (DELETE /deliveries/:id)

* Requiere autenticación y permisos

**Módulo de Laboratorios (/labs)**

1. **Reportar equipo dañado** (POST /labs/report/:rentalContractId)

* Requiere autenticación
* Parámetros:

      {

        "notes": "string"

      }

1. **Reportar equipo para reparación** (PATCH /labs/:id/repair)

* Requiere autenticación y rol de técnico
* Permite subir archivos de evidencia

**Módulo de Contratos (/contracts)**

1. **Obtener contratos del usuario** (GET /contracts/my)

* Requiere autenticación
* Retorna lista de contratos del usuario actual

**3. Implementación de Seguridad**

**Autenticación**

* Uso de JWT para tokens de autenticación
* Los tokens incluyen ID de usuario y roles
* Validación de tokens en cada petición protegida
* Se maneja expiración de tokens

**Autorización**

* Sistema de roles basado en JWT
* Guards para proteger rutas según roles
* Validación de permisos por endpoint
* Manejo de acceso basado en roles

**Persistencia**

* Base de datos PostgreSQL
* TypeORM como ORM

**Validación de Datos**

* DTOs para validación de entrada
* Manejo de errores
* Respuestas HTTP apropiadas