



# SISTEMA DE MONITORIZACIÓN DE SENSORES IOT

Monitor de sensores IoT para entornos de Living  
Lab

## Descripción breve

Este documento detalla el desarrollo e implementación de un sistema de monitorización de sensores IoT para entornos de Living Lab, construido utilizando el stack MERN (MongoDB, Express.js, React y Node.js).

ALFONSO SEBASTIÁN SIMORTE RAMOS  
KEVIN ALFONSO GÓMEZ SANDOVAL  
PAULINO ESTEBAN BERMÚDEZ RODRÍGUEZ

UCLM TALAVERA DE LA REINA | UBICUOS | 2024-2025

# Documentación Técnica y Manual de Usuario

## Resumen del proyecto

Este documento detalla el desarrollo e implementación de un sistema de monitorización de sensores IoT para entornos de Living Lab, construido utilizando el stack MERN (MongoDB, Express.js, React y Node.js). El proyecto se desarrolló siguiendo la metodología Scrum, enfatizando el desarrollo iterativo y la mejora continua.

## 1. Descripción General del Proyecto

El sistema de monitorización de sensores IoT facilita la gestión y análisis en tiempo real de datos dentro de un Living Lab. Ofrece autenticación intuitiva, un panel de control responsive con visualizaciones claras del estado de sensores activos, y herramientas para identificar patrones y anomalías. Permite gestionar dispositivos de forma sencilla, incluyendo detalles técnicos, ubicación e historial de mediciones, optimizando el mantenimiento y la comprensión del entorno.

### 1.1 Características Principales

- Monitorización de datos de sensores en tiempo real
- Autenticación de usuarios y control de acceso basado en roles
- Gestión y configuración de dispositivos
- Análisis y visualización de datos históricos
- Sistema de alertas automatizado
- Registro de auditoría para actividades del sistema

### 1.2 Stack Tecnológico

- **Frontend:** React con framework Next.js
- **Backend:** Node.js con Express.js
- **Base de datos:** MongoDB
- **Autenticación:** Sistema basado en JWT
- **Comunicaciones en tiempo real:** Implementación WebSocket

## 2. Arquitectura del Sistema General

El sistema utiliza una arquitectura moderna con frontend en React y Next.js para una experiencia fluida, y backend en Node.js con Express y MongoDB para la lógica de negocio y

almacenamiento. Incorpora un sistema de roles y permisos robusto que controla el acceso a funcionalidades, permitiendo a los administradores gestionar usuarios y monitorizar actividades con un sistema de auditoría. Los usuarios regulares pueden acceder a herramientas como visualización de datos e informes, aunque este módulo aún está en desarrollo parcial.

## 2.1 Arquitectura Backend

El backend implementa una estructura modular con:

- Capa de modelos para la representación de datos (Usuario, Sensor, DatosSensor, etc.)
- Capa de rutas para endpoints API
- Middleware de seguridad para autenticación y autorización
- Servicios auxiliares para auditoría y seguridad

### *Arquitectura del Sistema y Módulos Principales*

La plataforma se fundamenta en una arquitectura modular cuidadosamente diseñada para gestionar el ecosistema de sensores del Living Lab. El núcleo del sistema, implementado en el archivo `app.js`, establece la configuración esencial y la conexión con la base de datos MongoDB, asegurando una comunicación fluida entre todos los componentes. Este módulo central implementa medidas de seguridad robustas y gestiona las rutas principales que dan vida a la aplicación.

### *Gestión de Usuarios y Seguridad*

El sistema incorpora un modelo avanzado de gestión de usuarios y seguridad. Utiliza el módulo `User.js` para autenticación segura con cifrado de contraseñas y registro de información básica, gestionando automáticamente datos como fecha de creación y roles asignados. Los módulos `Role.js` y `Permission.js` refuerzan la seguridad, controlando detalladamente los permisos y acciones disponibles para cada usuario.

### *Control y Monitorización de Dispositivos*

La gestión de dispositivos utiliza los módulos `Sensor.js` y `Camera.js`, que almacenan detalles como ubicación, IP, firmware y estado de conexión. El módulo `SensorData.js` administra las mediciones, incluyendo ubicación, valor, unidad y marca temporal. Estos datos se agrupan en sesiones mediante `Session.js`, optimizando el análisis posterior.

### *Auditoría y Trazabilidad*

El sistema mantiene un registro detallado de todas las actividades a través del módulo `Audit.js`. Este componente registra cada acción significativa realizada en el sistema, almacenando

información sobre el usuario que la realizó, la fecha y la naturaleza de la acción. Esta capacidad de auditoría resulta fundamental para el mantenimiento de la seguridad y el cumplimiento de requisitos de trazabilidad.

#### *Gestión de Sesiones y Grabaciones*

La plataforma organiza la recopilación de datos en sesiones con el módulo *Session.js*, vinculadas a usuarios específicos e integrando múltiples dispositivos, con marcas temporales de inicio y fin para facilitar el análisis. El módulo *Record.js* gestiona las grabaciones de cámaras, registrando detalles como nombre de archivo, formato, duración, tamaño y marcas temporales, optimizando estudios visuales o de comportamiento en el Living Lab.

#### *Sistema de Enrutamiento*

El sistema de enrutamiento constituye el núcleo de la comunicación entre el cliente y el servidor, proporcionando una estructura organizada para la gestión de las diferentes funcionalidades del sistema. La implementación se ha realizado mediante *Express.js*, permitiendo una gestión eficiente y segura de las peticiones HTTP.

#### *Gestión de Dispositivos*

El módulo de enrutamiento unifica la gestión de sensores y cámaras mediante endpoints que soportan operaciones CRUD. Utiliza un filtrado inteligente para identificar el tipo de dispositivo y responder adecuadamente, permitiendo una fácil expansión a nuevos dispositivos sin alterar la estructura base.

#### *Gestión de Sesiones*

El sistema de sesiones organiza y gestiona mediciones en períodos específicos, permitiendo crear, consultar y administrar sesiones que pueden incluir múltiples dispositivos. Incorpora un sistema de paginación que optimiza el rendimiento al limitar por defecto a 50 registros por petición, facilitando la recuperación eficiente de grandes volúmenes de datos.

#### *Autenticación y Control de Usuarios*

El módulo de usuarios utiliza autenticación basada en JWT, gestionando registro, inicio de sesión y validación de tokens de acceso. Implementa validaciones robustas para garantizar la integridad de los datos, como la verificación de correos electrónicos duplicados, y asegura contraseñas mediante técnicas de hash.

#### *Gestión de Roles y Permisos*

El sistema de roles ofrece una autorización flexible, permitiendo definir y gestionar niveles de acceso mediante la asignación de permisos específicos a cada rol. Su diseño facilita una gestión

granular de las capacidades de los usuarios y admite la expansión o modificación de roles según las necesidades organizativas.

## 2.2 Arquitectura Frontend

El frontend utiliza Next.js con:

- Sistema de enrutamiento basado en páginas
- Arquitectura basada en componentes
- Gestión de estado basada en contextos
- Rutas protegidas con autenticación
- Interfaces de panel de control responsivas

### *Sistema de Enrutamiento*

El sistema de enrutamiento, implementado con Express.js, gestiona de manera eficiente y segura las peticiones HTTP, sirviendo como el núcleo de la comunicación entre el cliente y el servidor y proporcionando una estructura organizada para las diferentes funcionalidades del sistema.

### *Gestión de Dispositivos*

El módulo de enrutamiento de dispositivos gestiona de manera unificada sensores y cámaras, permitiendo operaciones CRUD a través de endpoints específicos. La ruta principal incluye un sistema de filtrado inteligente que identifica el tipo de dispositivo solicitado y proporciona la información correspondiente. Este enfoque flexible facilita la expansión futura a nuevos dispositivos sin modificar la estructura base del sistema.

### *Autenticación y Control de Usuarios*

El módulo de usuarios implementa un sistema completo de autenticación basado en JWT (JSON Web Tokens). Este componente gestiona el registro de nuevos usuarios, el inicio de sesión y la validación de tokens de acceso. El sistema incorpora validaciones robustas para asegurar la integridad de los datos de usuario, incluyendo la verificación de correos electrónicos duplicados y la gestión segura de contraseñas mediante técnicas de hash.

## Servicios Auxiliares

### *Sistema de Auditoría*

El servicio de auditoría implementa un mecanismo automatizado para el registro de acciones significativas en el sistema. Cada registro de auditoría incluye información detallada sobre el usuario que realiza la acción, la naturaleza de la acción y la marca temporal correspondiente. El

sistema ajusta automáticamente las zonas horarias para asegurar la precisión en el registro temporal de las acciones.

### *Seguridad y Validación*

El módulo de seguridad proporciona una capa de protección mediante la validación de tokens JWT. Este componente verifica la autenticidad y validez de los tokens de acceso en cada petición protegida, asegurando que solo los usuarios autorizados puedan acceder a funcionalidades específicas. La implementación incluye manejo de errores robusto y mensajes claros para facilitar la identificación y resolución de problemas de autenticación.

### *Gestión de Grabaciones*

El sistema incluye un módulo especializado para la gestión de grabaciones, particularmente relevante para los dispositivos de tipo cámara. Este componente permite la recuperación y gestión de registros de grabación, incluyendo información sobre duración, formato y ubicación de los archivos. La implementación facilita la asociación entre grabaciones y dispositivos específicos, permitiendo un seguimiento detallado de las actividades de monitorización visual.

## 3. Manual de Usuario

### 3.1 Primeros Pasos

Para acceder al sistema:

1. Navegar a la página de inicio de sesión url de inicio de sesión: <http://localhost:3000>

### Iniciar Sesión

Usuario \*

Pau

Contraseña \*

....

INICIAR SESIÓN

¿No tienes una cuenta? [Regístrate aquí](#)

2. Introducir las credenciales para un nuevo usuario en la URL `http://localhost:3000/register`

## Regístrate

Formulario de registro:

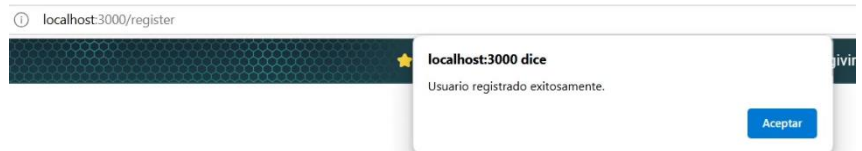
Usuario: alex

Correo: alex@gmail.com

Nombre Completo: alex

Contraseña: ....

Botón: REGISTRARSE



3. Tras la autenticación exitosa, será dirigido al panel de control

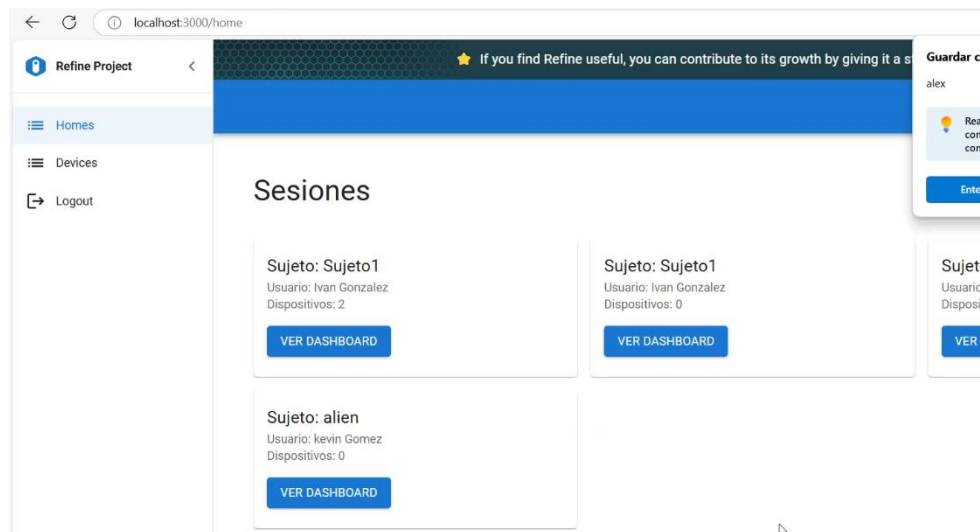
## Iniciar Sesión

Formulario de inicio de sesión:

Usuario \*: alex

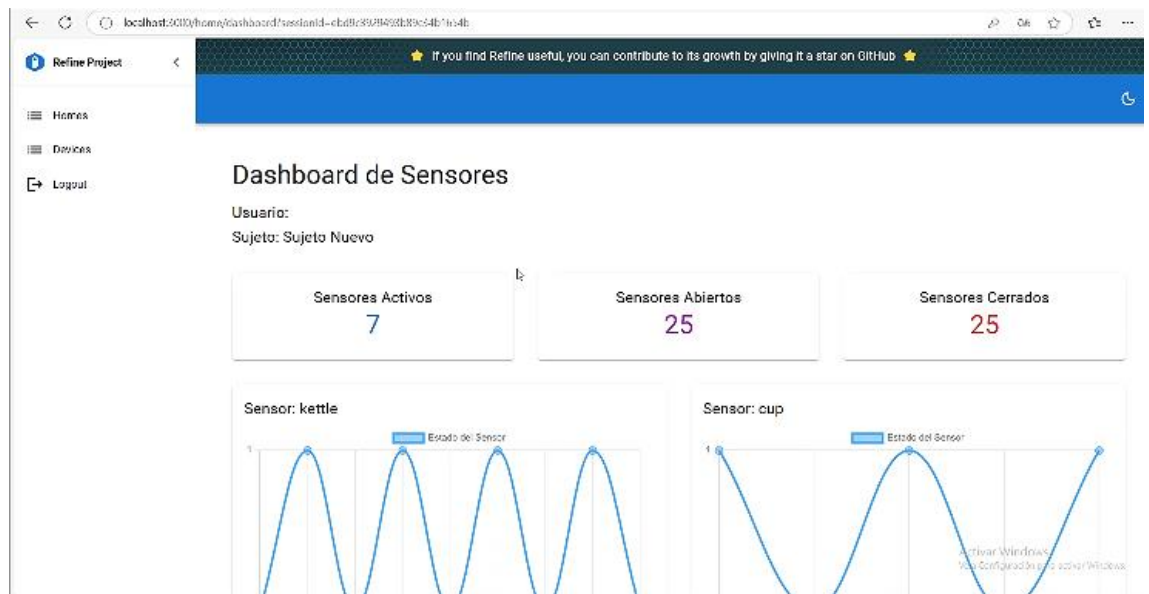
Contraseña \*: ....

Botón: INICIAR SESIÓN



¿No tienes una cuenta? [Regístrate aquí](#)

4. Visualización de los datos de los sensores. Clic en el Sujeto: Sujeto Nuevo (En la actual demo es quien posee los datos a visualizar)



## 3.2 Vista General del Panel de Control

El panel de control proporciona:

- Visualización de lecturas de sensores en tiempo real
- Indicadores de estado de dispositivos
- Acceso rápido a funciones frecuentemente utilizadas
- Alertas y notificaciones del sistema

## 3.3 Gestión de Dispositivos

Para gestionar sensores:

1. Navegar a la sección de Dispositivos
2. Ver todos los dispositivos registrados
3. Añadir nuevos dispositivos usando el botón "Crear"
4. Editar configuraciones de dispositivos existentes según sea necesario
5. Monitorizar el estado y la salud de los dispositivos

## 3.4 Análisis de Datos

El sistema ofrece varias capacidades de análisis de datos:

- Visualización de datos históricos
- Análisis de tendencias
- Generación de informes personalizados



- Funcionalidad de exportación de datos

## 4. Detalles de Implementación Técnica

### 4.1 Esquema de Base de Datos

- **La base de datos MongoDB** implementa las siguientes colecciones principales:
- **Usuarios:** Gestión de usuarios y autenticación
- **Sensores:** Configuración de dispositivos y metadatos
- **DatosSensor:** Datos de series temporales de sensores
- **Auditoría:** Registro de actividad del sistema
- **Sesiones:** Gestión de sesiones de usuario

### 4.2 Estructura API

La API backend está organizada en rutas lógicas:

- **/devices:** Gestión de dispositivos sensores
- **/users:** Administración de usuarios
- **/records:** Registros de datos de sensores
- **/sessions:** Gestión de sesiones
- **/roles:** Control de acceso basado en roles

### 4.3 Implementación de Seguridad

Las medidas de seguridad incluyen:

- Autenticación basada en JWT
- Control de acceso basado en roles
- Protección de rutas API
- Registro de auditoría
- Gestión de sesiones

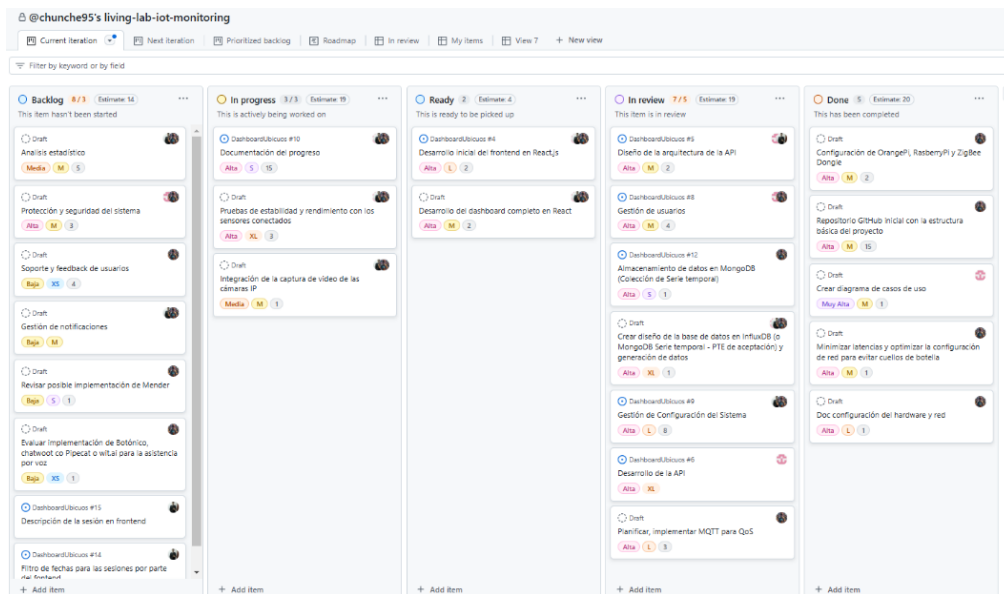
## 5. Proceso de Desarrollo

### 5.1 Implementación Scrum

El proyecto se desarrolló utilizando la metodología Scrum con:

- Ciclos de sprint de dos semanas
- Reuniones diarias de seguimiento

- Planificación de sprint y retrospectivas



## 5.2 Historial de Sprints

Logros notables a través de los sprints:

- **Sprint 1:** Autenticación básica y gestión de usuarios bajo el rol de Usuario estándar.
- **Sprint 2:** Recolección y almacenamiento de datos de sensores
- **Sprint 3:** Implementación de monitorización en tiempo real
- **Sprint 4:** Características de panel de control y visualización
- **Sprint 5:** Análisis avanzado y generación de informes

## 6. Otros

Se ha desarrollado el proyecto paralelo de “justmovedoc” para la retroalimentación del manual de usuario, versión extendida.

## 7. Conclusión

Este sistema de monitorización de sensores IoT proporciona una base robusta para la gestión de Living Labs, combinando capacidades de monitorización en tiempo real con herramientas completas de análisis de datos. La arquitectura modular del sistema y el código base bien documentado aseguran la mantenibilidad y extensibilidad para mejoras futuras.

**Versión del Documento:** 1.2.5

**Última Actualización:** 17 de enero de 2025