

# Visualización de datos en tiempo real (Angular + Node + Socket.IO + MongoDB)

## Requisitos

- Node.js: 20.19.4
- npm: 10.x
- Angular CLI: 20.x
- MongoDB Atlas (cadena de conexión válida)

## Estructura del repositorio

```
Back/  
  src/  
    index.js  
    routes/  
      info.js  
    models/  
      info.js  
    scripts/  
      simulator.mjs  
  .env                # Variables de entorno (no commitear)  
Front/  
  ... (aplicación Angular)
```

## Variables de entorno (Back/.env)

```
MONGO_URI=<tu-uri-de-atlas>  
PORT=4000  
CORS_ORIGIN=http://localhost:4200
```

## Instalación

```
# Backend  
cd Back  
npm install  
  
# Frontend  
cd ../Front  
npm install
```

## Ejecución

```
# 1) Backend (HTTP + WebSocket en el MISMO puerto)  
cd Back  
node src/index.js  
# → HTTP+WS escuchando en http://localhost:4000  
  
# 2) Simulador de dispositivo (emite cada 10 s)  
cd Back  
npm run simulate  
  
# 3) Frontend (Angular)  
cd Front  
npx ng serve -o
```

# → abre http://localhost:4200

## WebSocket

Cliente → servidor (evento "iot/sensors"):

```
{
  "date": "ISO-8601",
  "data": [ { "value": <TEMP 10-25> }, { "value": <HUM 30-70> } ]
}
```

Servidor → clientes (evento "iot/sensores"): reenvía el mismo payload.

URL del socket en el Front: http://localhost:4000

## API REST

### POST /api/info

Crea una medición.

Body:

```
{ "sensor": "TEMP|HUM", "value": 21.5, "date": "2025-08-20T17:00:00.000Z" }
```

Validaciones:

- sensor debe ser TEMP o HUM
- Rangos permitidos: TEMP (10–25), HUM (30–70)

Respuestas: 201 creado, 400 inválido, 422 fuera de rango

```
curl -X POST http://localhost:4000/api/info -H "Content-Type: application/json" -d '{"sen
```

### GET /api/info

Consulta mediciones con filtros y paginación.

Parámetros soportados:

- minutes=15 → últimos N minutos
- sensor=TEMP|HUM + from=ISO + to=ISO
- limit (default 1000, máx 10000), skip
- aggregate=avg1m (opcional) → promedio por minuto

# Últimos 15 minutos (hasta 1000 docs)

```
curl "http://localhost:4000/api/info?minutes=15&limit=1000"
```

# Rango por sensor

```
curl "http://localhost:4000/api/info?sensor=TEMP&from=2025-01-01T00:00:00Z&to=2025-01-01T01:00:00Z"
```

# Agregación: promedio por minuto (últimos 60 min de HUM)

```
curl "http://localhost:4000/api/info?minutes=60&sensor=HUM&aggregate=avg1m"
```

## Frontend (Angular)

- Gráficas en tiempo real (Socket.IO) para Temperatura y Humedad.
- Vista Historial con selector (15/30/60/120/300 min) que consume GET /api/info?minutes=<N>.
- Dependencias clave usadas: @angular/\* 20.2.0, @angular/material 20.2.0, @angular/cdk 20.2.0, typescript 5.8.3

## Solución de problemas

- Socket conecta a :3000 → configurar el Front a http://localhost:4000 en SocketIoModule.forRoot(...).
- CORS bloqueado → revisar CORS\_ORIGIN en .env y reiniciar el Back.
- Fechas antiguas → el modelo guarda date como ISO; usa el parámetro minutes para ver datos recientes.