## CLIENTE

ΠY

# logica (proxy) obtenerTodasLasMediciones() [Medicion] <= posicion1:Posicion, posicion2:Posicion, t inicio:Z, final:Z, tipo:Z => obtenerMedicionesAcotadas() [Medicion] <= medicion:Medicion => crearMedicion() => usuario:Usuario => crearUsuario() => usuario:Usuario => actualizarUsuario() => id:Z => obtenerUsuario() Usuario <= nodo:Nodo => actualizarNodo() => nodo:Nodo => crearNodo() => id:Z => obtenerNodo() Nodo <= obtenerTodosLosNodos()

# ${\tt UsuariosControlador}$

- + ID:Z
- + telefono:Z
- + nombre:Texto
- + password:Texto
- + toString() -> json:Texto
- + getNombre() -> nombre:Texto

# NodosControlador

- + ID:Z
- + usuario:Z
- + estado:Z

### MedicionesControlador

- + ID:Z
- + nodo:Z
- + tiempo:Z (milis)
- + tipo:Z
- + posicion:Posicion
- + valor:R
- + usuario:Z

toString() -> json:Texto

getPosicion() -> posicion:Posicion

getTipo() -> tipo:texto

# API

### rutas

- + /medicion
- + /medicion/acotadas
- + /nodo
- + /medicion/all
- + /usuario

## logica

SERVIDOR

obtenerTodasLasMediciones()

### [Medicion] <=

posicion1:Posicion,

- posicion2:Posicion,
- t\_inicio:Z, t final:Z,
- tipo:Z =>

# obtenerMedicionesAcotadas() [Medicion] <=</pre>

medicion:Medicion =>

# crearMedicion() =>

usuario:Usuario =>

### crearUsuario() =>

usuario:Usuario =>

actualizarUsuario() =>

# id:Z =>

obtenerUsuario()
Usuario <=

### nodo:Nodo =>

actualizarNodo() =>
nodo:Nodo =>

### crearNodo() =>

id:Z =>
 obtenerNodo()

### Nodo <=

obtenerTodosLosNodos()

[Nodo] <=

# Posicion

- + latitud:R
- + longitud:R
- + getLatitud() -> lati
- + getLongitud() -> lone

### Usuario

- + ID:Z
- + telefono:Z
- + nombre:Texto
- + password:Texto
- + toString() -> json:Texto
- + getNombre() -> nombre:Texto

### Nodo

- + ID:Z
- + usuario:Z
- + estado:Z

### Punto

- + latitud:R
- + longitud:R
- + getLatitud() -> lati
- + getLongitud() -> long
- + distancia() -> R

m1\_database

### MedicionesControlador

- + ID:Z
- + nodo:Z
- + tiempo:Z (milis)
- + tipo:Z
- + posicion:Posicion
- + valor:R
- + usuario:Z

### toString() -> json:Texto

mediciones:[Texto] ->

interpolarMediciones()

mediciones <-

getPosicion() -> posicion:Posicion

getTipo() -> tipo:texto