**FIGURAS DE ESTRUCTURA:**

* Estructura principal: mostrar las bases de datos en esa figura
* Estructura frontend: repetir si se puede, nodes\_modules no tiene nada, fingir que tiene elementos añadiendo ficheros
* Estructura backend: quitar las bbdd de esa figura y que se vean las clases de configuración si eso

**FIGURAS DE LA BBDD:**

* Figura de la clase que contiene la tabla *AdditionalWatermeterData* (se necesita tabla y sentencia)

**FIGURAS DEL FRONTEND:**

* Figura para el código HTML: *watermeter.component.html* (que se vea el traductor transloco, los \*ngIf, los componentes hijos)
* Figura para TS: *watermeter-measures.component.ts* (se tiene que ver los datos para la comunicación de los hijos, el constructor y posiblemente un ejemplo de llamada a un servicio, que se vea el @Component)
* Figura para TS: *watermeter.component.ts* (para que tenga cierta coherencia con el html y poder enseñar los métodos de un servicio y la parte de comunicación con el hijo)
* Figura para service.ts: *watermeter.service.ts* (si se añade esta figura, puede que no haga falta la del watermeter.component.ts)
* Figura model.ts: *watermeter.ts* (para mostrar un modelo)

FIGURAS DEL BACKEND:

* Figura para model.java: *Watermeter.java* (para hacer la comparación con el modelo del front)
* Figura del repository.java: *AdditionalWatermeterDataRepository.java* (se puede buscar otra clase, la que parezca más interesante y au)
* Figura del controller: *WatermeterController.java* (donde se ven las peticiones get y todo eso, enseñar fragmento donde se muestre get y put)
* Figura de la configuración: *AppClientConfig.java* (mostrar la config de uno de los microservicios, incluyendo la asociación de @value y @bean)
* Figura del servicio: *AlarmService.java* (éste es un buen ejemplo, que se vean las sentencias lambda)