Hola,

para que vayas avanzando te comento:

el sistema que había estaba basado en MySQL. Después de verano me pasarán lo que tenían (espero...). Una decisión que tendrás que tomar es valorar si es mejor usar una base de datos u otra.

Por otro lado, como alternativa a una instalación propia de MySQL he investigado un poco con ThingSpeak y he encontrado esto que es justamente lo que queremos hacer <https://aprendiendoarduino.wordpress.com/tag/callback-sigfox/>. Thingspeak es un servicio de Mathworks (matlab) para ofrecer pequeñas bases de datos para dispositivos IoT y ofrecer un sistema muy básico de visualización y conexión con matlab para análisis, que es de donde ellos sacarán el dinero... Puedes hacerte una cuenta gratis indicando que es para uso personal o estudiantil <https://thingspeak.com/>. No es una base de datos tipo SQL como la otra, sino de más alto nivel que se configura en base a canales (una especie de tabla). Yo he creado dos canales:

1. <https://thingspeak.com/channels/1833561> tiene un único campo para almacenar toda la trama de sigfox, que se llama "frame":  
   Interfaz de usuario gráfica, Texto, Aplicación

   Descripción generada automáticamente  
   Y por defecto tiene una clave de lectura y una escritura. Estas claves son un código alfanumérico que tienes que añadir a cada petición para que la tarea se lleve a cabo. Por ejemplo en nuestro caso, la de escritura sólo sería usada por sigfox para reenviar los datos de los nodos aquí. Y la de lectura sería usada por tu página web para recibir los datos en función de los permisos que le pongas. Yo las he puesto todo público, así que para leer no hace falta nada.  
   Además, si te fijas en la siguiente imagen, también he dejado en la parte de la derecha unos ejemplos de peticiones de escritura y lectura que da la aplicación, en la sección de "API Requests":  
   Interfaz de usuario gráfica, Texto, Aplicación

   Descripción generada automáticamente  
   Estas peticiones son el método de HTTP, en este caso sólo usa GET (<https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto#GET>) y luego una dirección como si fuera una web (<https://api.things>...) y finalmente parámetros después del "?" (api\_key=lo de la izquierda, field1=el valor que voy a almacenar). Y podemos probarlo copiando el texto y llevándolo a una aplicación como curl o Advanced REST client (que es la que yo suelo usar). En la siguiente imagen indico el método, la dirección de antes y ya está!. Devuelve un código 200 que es que todo ha ido bien (<https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto#C%C3%B3digos_de_respuesta>)  
   Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

   Descripción generada automáticamente  
   Y finalmente puedes volver a la web de thingspeak para ver que los datos se han almacenado bien:  
   Interfaz de usuario gráfica, Aplicación

   Descripción generada automáticamente
2. <https://thingspeak.com/channels/1833578>. He creado un segundo canal suponiendo que los datos ya llegan parseados, es decir, se han dividido en la trama original y tenemos cada valor en su campo correspondiente de temperatura, humedad, índice UV, etc.  
   Interfaz de usuario gráfica, Aplicación

   Descripción generada automáticamente  
   Y como antes, probamos a escribir algunos valores (lo que tendría que hacer sigfox), pero en este caso, tres campos a la vez:  
   Interfaz de usuario gráfica, Texto, Aplicación

   Descripción generada automáticamente

En tu trabajo, tanto si se almacena un campo como varios, tendrás que leer los datos. Viendo los otros ejemplos de peticiones REST del API de thingspeak tenemos esto:  
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Donde arriba tenemos el get a una URL que incluye channels, el número de mi canal, feed, etc. Y la respuesta está abajo en azul, el JSON {"channel..." "feeds... y por aquí los últimos valores"}. Esto es lo que tú tendrías que hacer desde una web para recibir los datos de thingspeak. También puede que no sólo haya que leer el channel, sino algo más como un campo en particular, el status del canal, añadir parámetros para indicar de cuánto tiempo quieres los datos, etc. La documentación del API de thingspeak está aquí: <https://www.mathworks.com/help/thingspeak/channels-and-charts-api.html>, la parte de REST (la mqtt es la que usarían los nodos, que no es nuestro caso). El formato JSON puede convertirse directamente en un objecto de javascript, así que el uso es bastante sencillo: <https://www.w3schools.com/js/js_json_parse.asp>.  A partir de ahí, ya es trabajar en la web la forma de visualización.

Creo que con esto tienes bastante por ahora para jugar con un REST API, intentar hacer peticiones primero con curl u otro cliente y luego llevar eso a una web.

Saludos.