

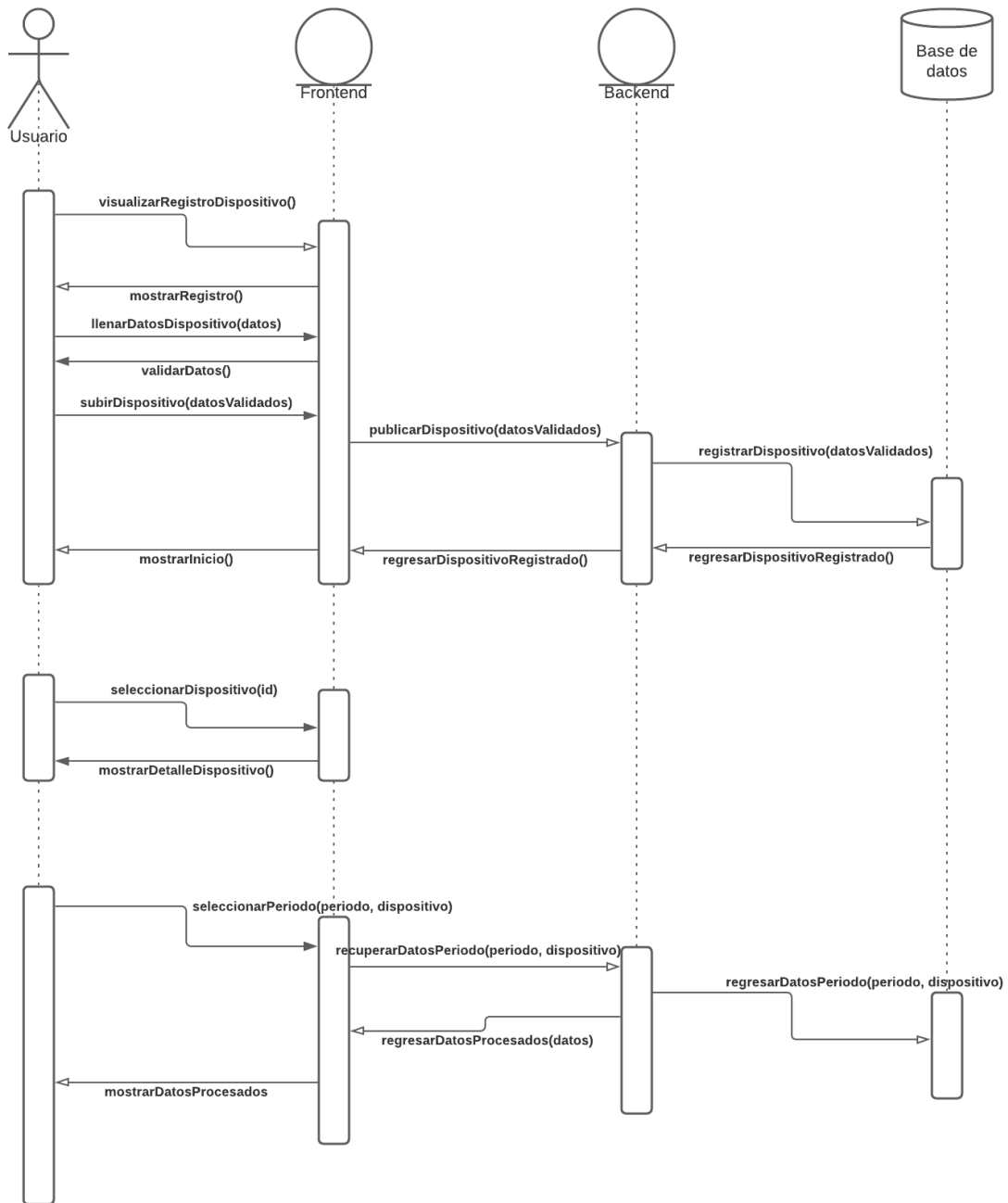
## Manual técnico

### *Patrones de diseño utilizados*

- Como parte del desarrollo realizado por equipos previos, se encontró que se empleó el patrón de diseño de Estado, esto debido a los siguientes puntos:
  - Se utilizó React para la implementación del cliente web.
  - React condiciona el comportamiento de sus componentes con base en el estado asignado a cada uno de ellos.
  - Cuando existe un cambio de estado de cierto componente, esto se traduce en un comportamiento de re-rendering, que permite modificar la información, apariencia, procesos, etc que se encontrarán presentes en la aplicación.
- También, de manera indirecta, se encontró el uso del patrón de Observador, por lo siguiente:
  - Se empleó la librería Redux en el cliente web.
  - Esta librería permite el control de un estado centralizado que condicione el comportamiento de todos aquellos componentes que se suscriban al mismo.
  - Debido a esto, cada vez que el estado centralizado es modificado, se le notifica a todos los componentes que están suscritos al mismo.

### *Diagrama de secuencia*

El siguiente diagrama muestra de manera sencilla la manera en la que el sistema muestra a los usuarios el consumo de un dispositivo en un periodo seleccionado.



### Modelo de base de datos

El modelo empleado es sumamente sencillo. Solamente se encontraron 3 entidades en la aplicación: Usuario, Dispositivo, Datos\_Consumo. De igual manera, se presentaron las siguientes relaciones:

