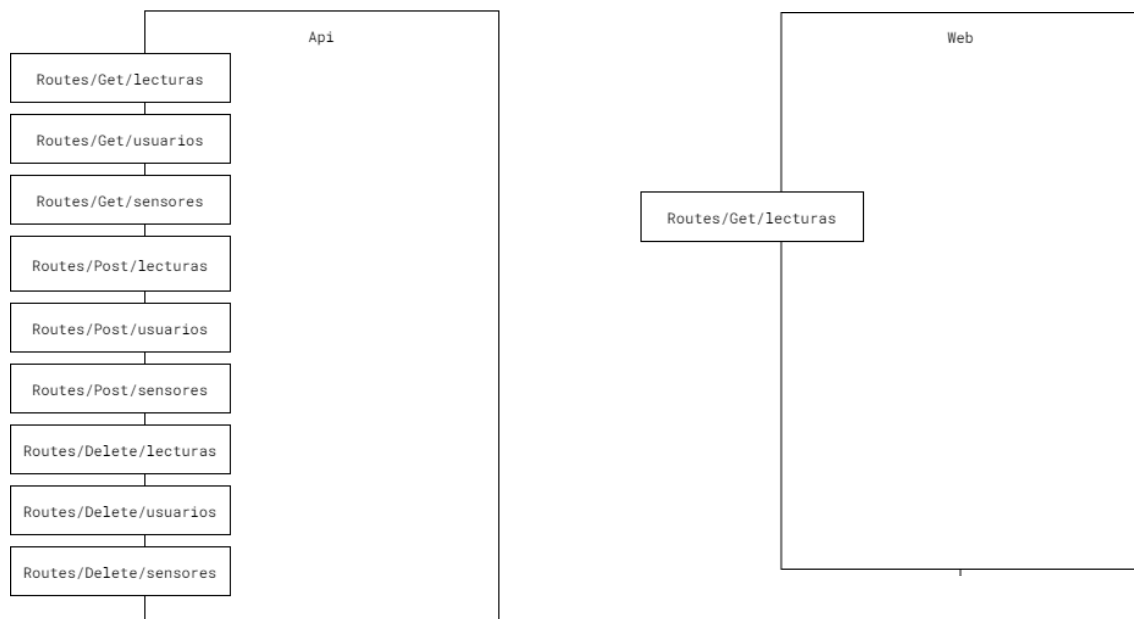


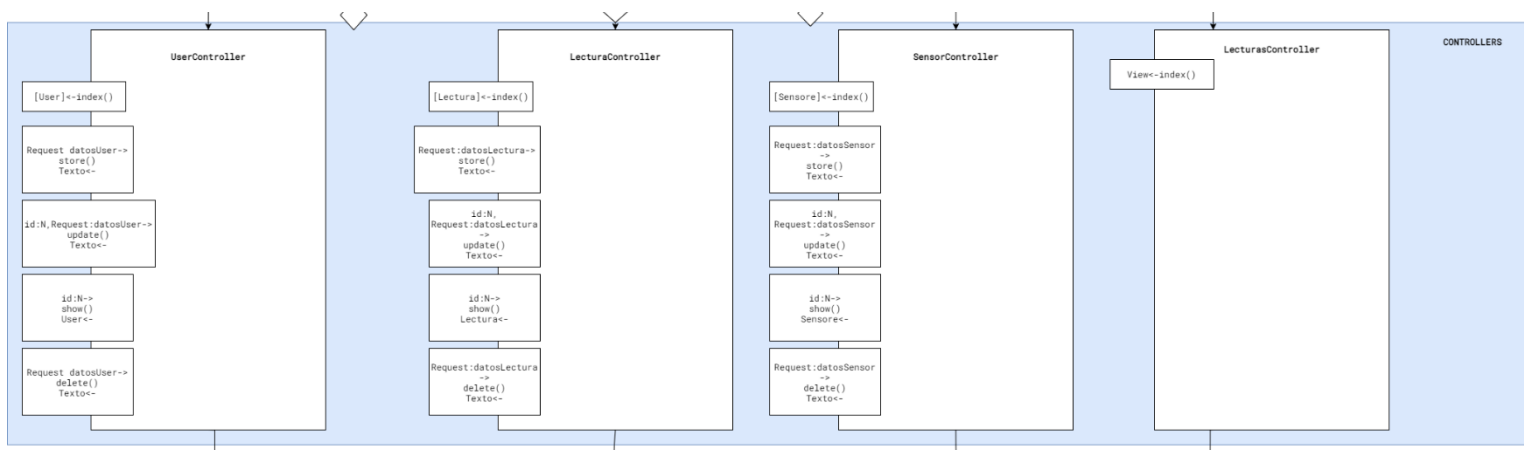
Documento de diseño Backend

Jose Julio Peñaranda Jara

A continuación, se muestra el diseño de los archivos de rutas con sus peticiones get post y delete. El archivo api se encarga de devolver y guardar datos y el archivo web se comunica con el frontend para mostrar archivos de la BBDD o hacer demás operaciones desde la pagina web



El archivo de rutas API llama a los respectivos controladores para ejecutar cada una de las rutas, existe un controlador para cada una de las columnas de la base de datos. También existe un archivo LecturasController al que llama el archivo web para mostrar todas las mediciones desde su método llamando a la lógica



```

graph TD
    subgraph LogicaNegocio
        direction TB
        L1["R:data, Fecha:time_data, N:id_sensor  
->guardarMedicion()  
Error|Texto<-"]
        L2["Texto:Name, Texto:Email, Texto:Password  
->guardarUser()  
Error|Texto<-"]
        L3["obtenerTodosLosUser()->[User]"]
        L4["N:id->obtenerUserConId()->User"]
        L5["N:id  
->eliminarUser()  
Error|Texto<-"]
        L6["N:numMedicines->  
obtenerUltimasMedicines()->[Lectura]"]
    end

    subgraph Logica
        direction TB
        L7["Texto:Name, Texto:Model,  
Texto:Description  
->guardarSensor()  
Error|Texto<-"]
        L8["obtenerTodosLosSensores()->[Sensor]"]
        L9["N:id->obtenerSensorConId()->Sensor"]
        L10["N:id  
->eliminarSensor()  
Error|Texto<-"]
        L11["obtenerTodasLasMedicines()->[Lectura]"]
        L12["N:id->obtenerLecturaConId()->Lectura"]
    end

    L1 --> L2
    L2 --> L3
    L3 --> L4
    L4 --> L5
    L5 --> L6
    L6 --> L7
    L7 --> L8
    L8 --> L9
    L9 --> L10
    L10 --> L11
    L11 --> L12

```

```

classDiagram
    class User {
        Texto:Name
        N:id
        Texto:Email
        Texto:Password
        User()->
    }
    class Lectura {
        N:id
        R:data
        Fecha:time_data
        N:id_sensor
        Lectura()->
    }
    class Sensores {
        N:id
        Texto:Name
        Texto:Description
        Texto:Model
        Sensores()->
    }
  
```

The diagram illustrates three entity classes: **User**, **Lectura**, and **Sensores**. Each class is represented by a large rectangle containing a list of attributes (smaller rectangles) and a constructor method (a rectangle with an arrow).

- User** attributes: Texto:Name, N:id, Texto:Email, Texto:Password. Constructor: User()->
- Lectura** attributes: N:id, R:data, Fecha:time_data, N:id_sensor. Constructor: Lectura()->
- Sensores** attributes: N:id, Texto:Name, Texto:Description, Texto:Model. Constructor: Sensores()->

A continuación, se muestra el diseño completo con las relaciones entre clases.

