

## Lignes de commandes :

Dans un premier temps, lors de la réalisation de la SAE201-204 consacré au développement d'une application en Java sous Raspberry Pie 3, nous avons du créé un utilisateur sur Rasbian, puis installer les différents packages nécessaires à notre programme.

Mise à jour de Rasbian

```
sudo apt update;  
sudo apt upgrade
```

Ceci fait, nous avons procédé à l'installation du Système de Gestion de Base de Données choisi au préalable : **MariaDB**.

```
sudo apt install mariadb-server;
```

Configuration de mariadb : définition du root-password

```
sudo mysql_secure_installation;
```

Se connecter à mysql

```
mysql -uroot -p;
```

Création de la base de donnée, des utilisateurs et de leur permissions.

```
Create database Sae;  
Create table sonsalle(id INT not null primary key auto_increment, temps  
VARCHAR(255), valSons VARCHAR(255), pourcentage VARCHAR(255), CONSTRAINT pk-  
sonsalle PRIMARY KEY(id));  
  
CREATE USER 'admin0' IDENTIFIED BY '0000';  
GRANT ALL PRIVILEGES ON *.* TO admin0 WITH GRANT OPTION;  
Flush privileges;  
  
CREATE USER 'superuser0' IDENTIFIED BY '0000';  
GRANT ALL PRIVILEGES ON *.* TO superuser0 WITH GRANT OPTION;  
Flush privileges;  
  
CREATE USER 'user0' IDENTIFIED BY '0000';  
GRANT CREATE, DELETE, SELECT, ALTER ON sonsalle.* TO user0;  
Flush privileges;
```

Installation de JDK8 pour le remote plateforme entre le Raspberry et nos ordinateurs personnels.

```
sudo apt install openjdk-8-jdk  
java -version
```

Récupérer l'adresse IP pour le remote-run

```
hostname -I
```

Au sein de l'application

Concernant la base de donnée de l'application en elle même, premièrement on a commencé par créé une connexion à une base de donnée quelconque issu de la base de donne EDT, afin de tester l'affichage des différents graphiques relatifs à celle-ci.

```
Connection conn = null;

    DefaultPieDataset dataset = new DefaultPieDataset();

    // driver
    Class.forName("com.mysql.cj.jdbc.Driver");
    // driver parameters
    String url = "jdbc:mysql://iutbg-lamp.univ-lyon1.fr:3306/EDT";
    String user = "p2100028";
    String password = "12100028";

    // create a connection to the database
    conn = DriverManager.getConnection(url, user, password);
```

Après avoir créé la connexion, on lis les données de la base de donnée, et on les affiche dans la console pour vérifier leurs enregistrements.

```
String strInsert = "SELECT * FROM Groupes";

stLogin = conn.createStatement();
rsLogin = stLogin.executeQuery(strInsert);

//affichage

        System.out.println("-----");
        System.out.println("Id groupe " +
rsLogin.getString("idGroupe"));
        System.out.println("effectif " +
rsLogin.getString("effectif"));
        System.out.println("type " + rsLogin.getString("type"));
        System.out.println("parent " + rsLogin.getString("parent"));
```

Puis on test les différentes propriétés de la librairie JFreeChart d'après des boutons (JMenuItem) placer en JMenuBar de notre applications.

Après avoir réussi à correctement tracer les graphiques qui nous intéressaient, on a créé la base de donnée exemple présenté plus tôt, sur le Raspberry, dont on à cherché ensuite à lire les données depuis notre ordinateur grâce au système de Java remote plateforme.

Ceci fini, on est passé aux objectifs restants en s'attaquant à celui concernant le capteur.

**Suite à un vote démocratique des plus complexe, et élu par la majorité absolue, c'est le sujet concernant le capteur sonore qui fut choisi.**

Après lecture des documentations du capteur Grove Sound Sensor, nous l'avons branché sur le module prévu à cette effet, sur l'un des ports Analogiques d'entrée, nous permettant ainsi la récupération des données.

Cependant, un problème survient, impossible de lire les données du capteur. Les librairies appelées, sencées nous retourner les valeurs de volume sonore ne renvoie que des "0".

Le problème fut réglé après quelques heures de recherche par l'activation du bus I2C sur le

Raspberry.

Notre problème étant réglé, nous avons donc pu récupérer les données non nulles du capteur, grâce à la commande :

```
String strInsert = "SELECT * FROM sonsalle"
Double.parseDouble(rsLogin.getString(3));
        if
(Integer.parseInt(rsLogin.getString("valSons")) > 0) {
            d.addValue(Double.parseDouble(rsLogin.getString(3)) / 100,
"valeur du son", rsLogin.getString(1));
        }
```

Puis on rajoute cette donnée à une variable 'd' qui correspond à un DefaultPieDataset :

```
                d.addValue(Double.parseDouble(rsLogin.getString(3)) / 100,
"valeur du son", rsLogin.getString(1));
```

Ensuite on insert ces données dans la table que nous avons créé juste avant :

```
String addValue = "INSERT INTO sonsalle(id,temps,valSons,pourcentage) values (
NULL," + temps + ",'" + valeurson + "','" + pourcentage + "')";
        stLogin.executeUpdate(addValue);
```

Puis ensuite, une fois qu'on a récupéré les valeurs du capteur et après les avoir insérés dans une base de donnée, on se sert de cette nouvelle base de donnée pour afficher des graph, mais aussi une fenêtre avec grille de données comme ceci :

```
String strInsert = "SELECT * FROM sonsalle";
        //String SQLeffectiftotal = "select SUM(effectif) from Groupes
where idGroupe like 'S_';

        //statement
        stLogin = conn.createStatement();
        rsLogin = stLogin.executeQuery(strInsert);

        model = (DefaultTableModel) jTable1.getModel();

model.addRow(ligne);
```

Mais aussi, avec cette fenêtre, il est possible de supprimer des éléments issus de cette base de donnée avec cette commande :

```

int numRowsSelected = jTable1.getSelectedRow();

        if(numRowsSelected == -1){
            JOptionPane.showMessageDialog(this, "Selectionner une ligne");
            return;
        }

        String id=jTable1.getValueAt(numRowsSelected, 0).toString();

        //supp de la base
        stLogin.executeUpdate("DELETE FROM sonsalle where id="+ id );

```

Ou en ajouter de cette façon :

```

stLogin.executeUpdate("INSERT INTO sonsalle (id,temps,valSons,pourcentage)
values (" + idUI.getText()+ "," + tempsUI.getText()+ "','"+
        + valUI.getText() + "','"+ pourUI.getText()+"');"");

```

Sur cette fenetre l est aussi possible de créé des utilisateurs comme ceci avec cette commande :

```

        stLogin.executeUpdate("CREATE USER " + nameUI.getText()+ " IDENTIFIED BY '" +
passwordUI.getText()+ "';");

        stLogin.executeUpdate("GRANT INSERT, UPDATE, SELECT PRIVILEGES
ON Sae.* TO '"+ nameUI.getText()+"' WITH GRANT OPTION;");

```

Puis les ajouter dans la table que nous avons créée spécialement pour les utilisateurs comme ceci :

```

stLogin.executeUpdate("INSERT INTO usertable (id,username,password,droit)
values (0,'" + nameUI.getText()+ "','"+ passwordUI.getText()+
        "','"+ choix +"');"");

```

Donc, sur cette fenetre il est possible de realiser plusieurs actions précises liées à la gestion de base de donnée.

**En dehors de ce problème de données de capteurs, aucun réel soucis ne s'est présenté à nous lors de cette SAÉ, si ce n'est quelques erreurs de compréhension des documentations et des énoncés ainsi que quelques fautes d'organisations, auxquels nous avons su répondre rapidement.**