



Ce document est l'un des livrables à fournir lors du dépôt de votre projet : 4 pages maximum (hors documentation).

Pour accéder à la liste complète des éléments à fournir, consultez la page [Préparer votre participation](#).

Vous avez des questions sur le concours ? Vous souhaitez des informations complémentaires pour déposer un projet ? Contactez-nous à info@trophees-nsi.fr.

NOM DU PROJET : SURVEILLANCE AQUAPONIE

➤ PRÉSENTATION GÉNÉRALE :

- *Idée et objectifs*
- *Origines et intérêts du projet*
- (...)

Le professeur M PLANCHAIS a proposé cette idée d'aquaponie connecté. 4 élèves ont acceptés ce défi en 5 semaines.

Nous avons pu constater qu'il n'existait pas de projets complets sur internet de solution « libre » pour connecter son système d'aquaponie.

L'idée est de proposer un guide pour faciliter cette installation. Cela s'adresse à deux types de personnes.

- Aux enseignants qui souhaitent promouvoir ce type de culture auprès des élèves. C'est un projet qui regroupe plusieurs disciplines (SVT, SPH, NSI, SNT...)
- Aux particuliers qui pratiquent l'aquaponie et qui souhaitent se lancer dans l'aventure du connecté.

Nous avons prévu 2 mode. Un mode tkinter pour ceux qui ne souhaitent pas installer de serveur web. Il est prévu pour fonctionner avec un écran raspberry à accrocher sur l'aquarium.

Un mode connecté est également proposé et devra être amélioré, les possibilités sont nombreuses.

Ce projet a permis de mettre en pratique les commandes Linux , de découvrir les codes php et de mieux comprendre le SGBD. Enfin, Travailler sur un projet concret est un bon lien vers le grand oral et facilite les entretiens pour la poursuite d'études.

> ORGANISATION DU TRAVAIL :

- *Présentation de l'équipe (prénom de chaque membre et rôle dans le projet)*
- *Répartition des tâches*
- *Organisation du travail (répartition par petits groupes, fréquence de réunions, travail en dehors de l'établissement scolaire, outils/logiciels utilisés pour la communication et le partage du code, etc.)*

4 élèves de terminale NSI composent l'équipe, Eléa, Claudine, Gabriel et Manutea. Le professeur distribuait les tâches en fonction de l'avancement du projet et du temps disponible des élèves.

Pour chaque objectif, il fallait faire des recherches sur Internet et les tester sur un raspberry pi. Un document word permettait de noter les codes qui fonctionnaient.

Le projet a mis 5 semaines à se concrétiser. Nous avons travaillé 1 séance par semaine en classe sur le projet, 2 samedi matin, et quelques heures le mercredi après-midi. Certains ont pu travailler également chez eux pour finaliser.

LES ÉTAPES DU PROJET :

- *Présenter les différentes étapes du projet (de l'idée jusqu'à la finalisation du projet)*

Le professeur a proposé ce projet a organisé les étapes :

- Renseignements sur aquaponie : Gabriel
- Schématisation du système : Eléa
- Interface Tkinter et test des capteurs et actionneurs : Manutea
- Récupération des données analogiques : Manutea et Claudine
- Installation et test MariaDB, Apache, phpMyAdmin, Php : Claudine et Gabriel
- Programme python d'écriture sur base de données : Claudine et Gabriel
- Html et css: Eléa et le professeur
- Php : Manutea et Claudine
- Planificateur de tâches Crontab : Eléa et Gabriel
- Finalisation du projet : Gabriel, Eléa, Manutea et Claudine

➤ FONCTIONNEMENT ET OPÉRATIONNALITÉ :

- *Avancement du projet (ce qui est terminé, en cours de réalisation, reste à faire)*
- *Approches mises en œuvre pour vérifier l'absence de bugs et s'assurer de la facilité d'utilisation du projet*
- *Difficultés rencontrées et solutions apportées*

La première phase du projet est terminée. Elle consistait à mettre en place les solutions pour surveiller le système d'aquaponie. À partir de l'année scolaire prochaine d'autres élèves vont tester le système en faisant pousser des plantes. Ce sera l'occasion de tester et fiabiliser le système.

En fonction de l'analyse des capteurs et de la croissance des plantes, des « mises en garde » pourront être installées sur le système. En fonction de la phase de croissance des plantes, le pH optimal peut changer par exemple.

Tkinter

Le projet Tkinter fonctionne correctement.

Interface web

Le projet actuel affiche la dernière ligne de la base de données. L'exécution d'un programme python permet l'ajout d'une ligne dans la base de données. L'ensemble fonctionne correctement.

- Il reste à mettre en place le planificateur de tâche crontab (les recherches n'ont pas abouti).
- Les recherches sur le contrôle des relais (led et pompe) n'ont pas abouti également

> OUVERTURE :

- *Idées d'améliorations (nouvelles fonctionnalités)*
- *Stratégie de diffusion pour toucher un large public (faites preuve d'originalité !)*
- *Analyse critique du résultat (si c'était à refaire, que changeriez-vous dans votre organisation, les fonctionnalités du projet et les choix techniques ?)*

Améliorations :

- D'un capteur de niveau d'eau sur le bac à plantes pour vérifier que l'eau est bien reversée dans l'aquarium.
- Contrôle à distance des actionneurs.
- Graphiques de l'évolutions des mesures prises par les capteurs
- Un autre projet serait de fabriquer un système autonome pour la nourriture des poissons

Pour le mode TKINTER : On pourrait ajouter facilement un bouton qui permet de télécharger un csv des données recueillies.

Stratégies de diffusion :

L'idée est

DOCUMENTATION

L'objectif de ce dossier est de permettre à un enseignant ou un particulier pratiquant déjà l'aquaponie d'installer le système en autonomie avec un minimum de compétences informatiques.

Ce dossier compile les recherches et les tests des élèves puis le résultat final.

1	Les capteurs numériques.....	1
1.1	Test Températures avec les sondes DS18B20.....	1
1.2	Test Camera AZ- Delivery.....	2
1.3	Test Module détecteur de lumière à LDR.....	3
2	Les actionneurs.....	4
2.1	Test Ruban led.....	4
3	Capteur analogique.....	6
3.1	Test Capteur de pH avec connecteur BNC et capteur TDS.....	6
4	Le Projet Final Tkinter.....	7
5	MariaDb / MySQL.....	8
5.1	Installation MariaDb _MySQL.....	8
5.2	Gestion de la base de données depuis Python _ phase de test.....	9
6	Installation du serveur Apache.....	11
7	Installation de PHP sur la Raspberry.....	12
8	Installation de PHPMyadmin.....	13
9	Projet final.....	14
9.1	Le fichier python.....	14
9.2	Le fichier php et css.....	15
9.3	Code php pour affecter des variables.....	15
9.4	Code html.....	16

1 Les capteurs numériques

1.1 Test Températures avec les sondes DS18B20

Source : <https://raspberry-lab.fr/Composants/Sonde-de-temperature-DS18B20-sur-Raspberry-Francais/>

Préambule

1. Ajouter les lignes suivantes au fichier qui se trouve dans /etc/modules :
 - Pour modifier le fichier qui se trouve dans /etc/modules, on tape donc :

```
sudo nano /etc/modules
```

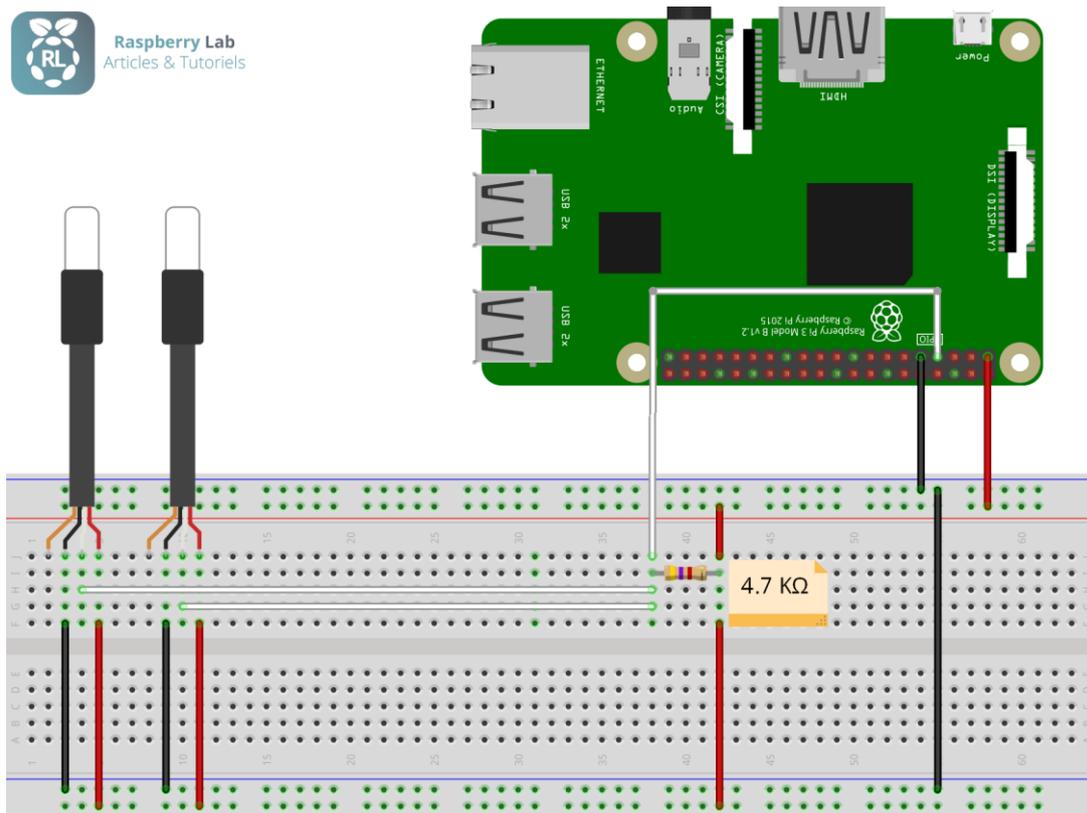
- ajouter

```
w1-therm  
w1-gpio pullup=1
```

2. Ajouter la ligne suivante au fichier qui se trouve dans /boot/config.txt :

```
dtoverlay=w1-gpio
```

Branchement



SOURCE : [HTTPS://RASPBERRY-LAB.FR/](https://raspberry-lab.fr/)

Fichier test python

```
import glob

def temperature():
    def lire_fichier (emplacement) :
        fichier = open(emplacement)
        contenu = fichier.read()
        fichier.close()
        return contenu

    def extraire_temperature (contenu) :
        seconde_ligne = contenu.split("\n")[1]
        donnees_temperature = seconde_ligne.split(" ")[9]
        return float(donnees_temperature[2:]) / 1000

routes_capteurs = glob.glob("/sys/bus/w1/devices/28*/w1_slave")

if len(routes_capteurs) > 0 :
    liste=[]
    c = 1
    for capteur in routes_capteurs :
        contenu_fichier = lire_fichier(capteur)
        temperature = extraire_temperature(contenu_fichier)
        liste.append(float(temperature))
        c += 1
else :
    liste=[0,0]
return liste
```

1.2 Test Camera AZ- Delivery

Source : <https://raspberrypi-lab.fr/Composants/Utilisation-Camera-sur-Raspberry-Pi-Francais/>

Préambule

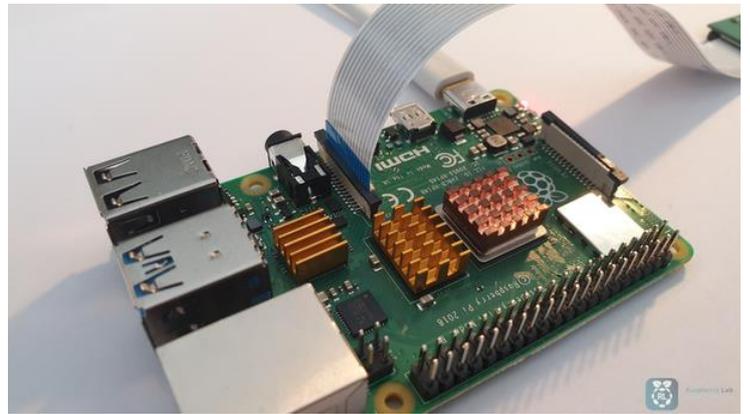
Dans un premier temps, il va falloir activer l'interface de la caméra. Entrez la commande suivante dans le terminal.

```
sudo raspi-config
```

Une fois dans le menu, **Interfacing Options** puis dans **Camera**. Enfin, sélectionnez **Yes** quand on vous demande la confirmation d'activation ainsi que pour redémarrer le Raspberry.

Branchement

Celle-ci dispose de son propre connecteur (*slot*). Il se trouve entre la prise Ethernet et la sortie HDMI.



SOURCE : [HTTPS://RASPBERRY-LAB.FR/](https://raspberrypi-lab.fr/)

Fichier test python

```
import picamera
import datetime

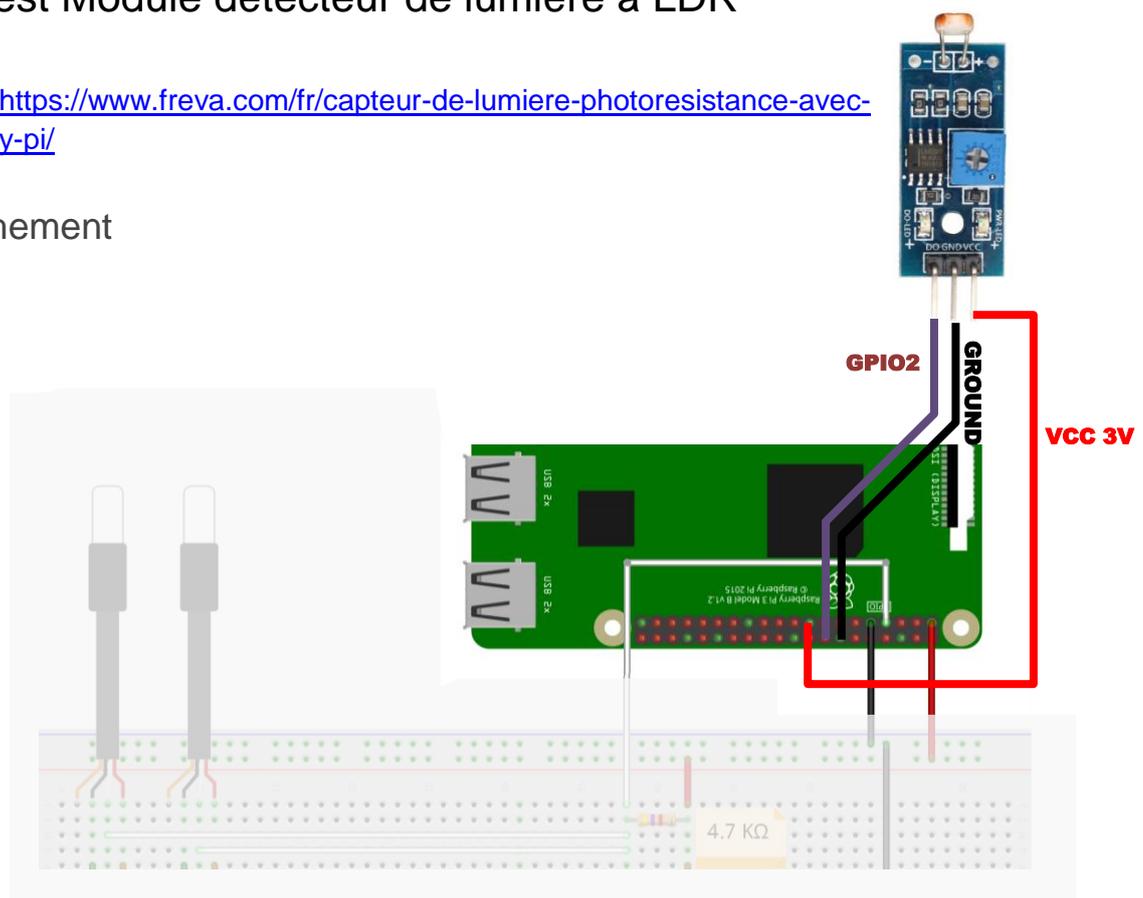
camera = picamera.PiCamera()
camera.resolution = (2592,1944)

#Enregistrement de la photo avec comme nom la date de prise de vue
camera.capture(str(datetime.datetime.now())+'.jpg')
```

1.3 Test Module détecteur de lumière à LDR

Source : <https://www.freva.com/fr/capteur-de-lumiere-photoresistance-avec-raspberry-pi/>

Branchement



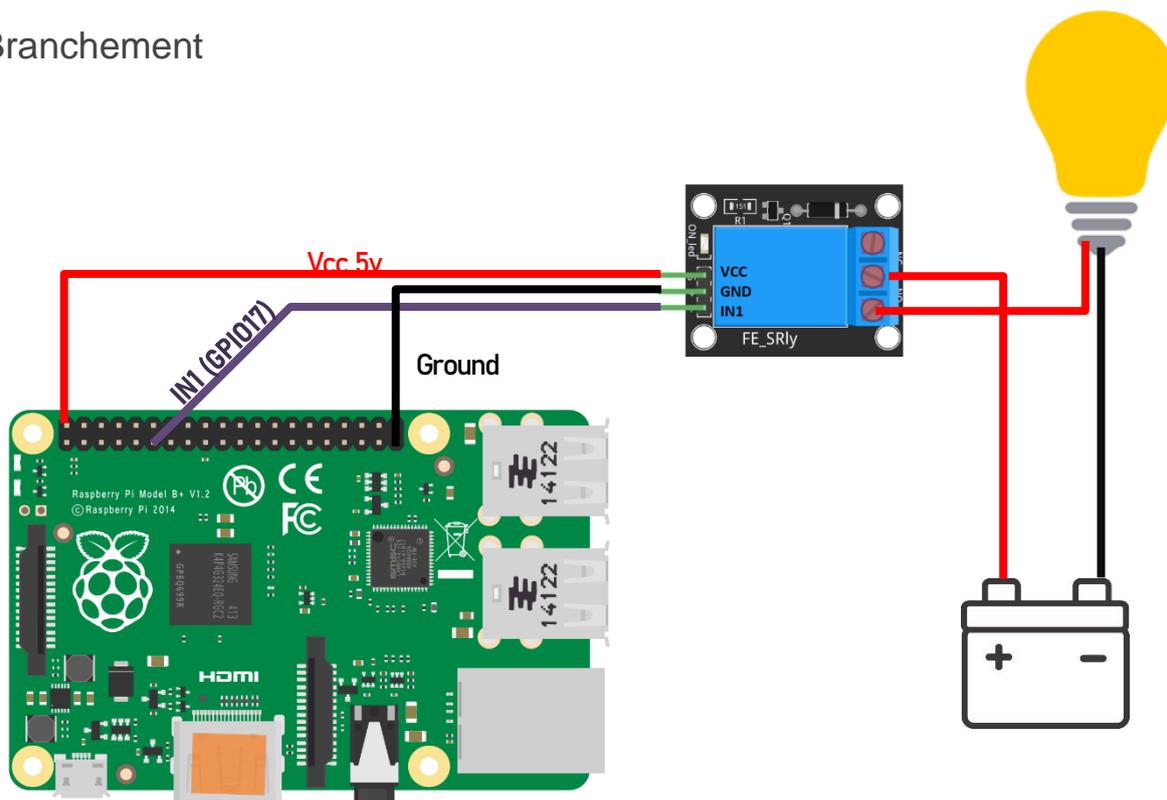
Fichier python

```
import RPi.GPIO as GPIO
import time
def luminosite():
    etat=0
    GPIO.setmode(GPIO.BCM)
    GPIO.setwarnings(False)
    LIGHT_PIN = 23
    GPIO.setup(LIGHT_PIN, GPIO.IN)
    IOld = not GPIO.input(LIGHT_PIN)
    IOld = GPIO.input(LIGHT_PIN)
    if GPIO.input(LIGHT_PIN):
        etat =0
    else :
        etat=1
    return etat
```

2 Les actionneurs

2.1 Test Ruban led

Branchement



Fichier test python

Source : <https://raspberrypi.pagesperso-orange.fr/dossiers/26-33.htm>

L'ampoule va clignoter, l'arrêt est prévu par un CTRL-C, le GPIO se désactive grâce à la fonction destroy()

```
import RPi.GPIO as GPIO
import time
RelayPin = 17 # pin17

def setup():
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(RelayPin, GPIO.OUT)
    GPIO.output(RelayPin, GPIO.HIGH)

def loop():
    while True:
        GPIO.output(RelayPin, GPIO.LOW)
        time.sleep(2)
        GPIO.output(RelayPin, GPIO.HIGH)
        time.sleep(2)

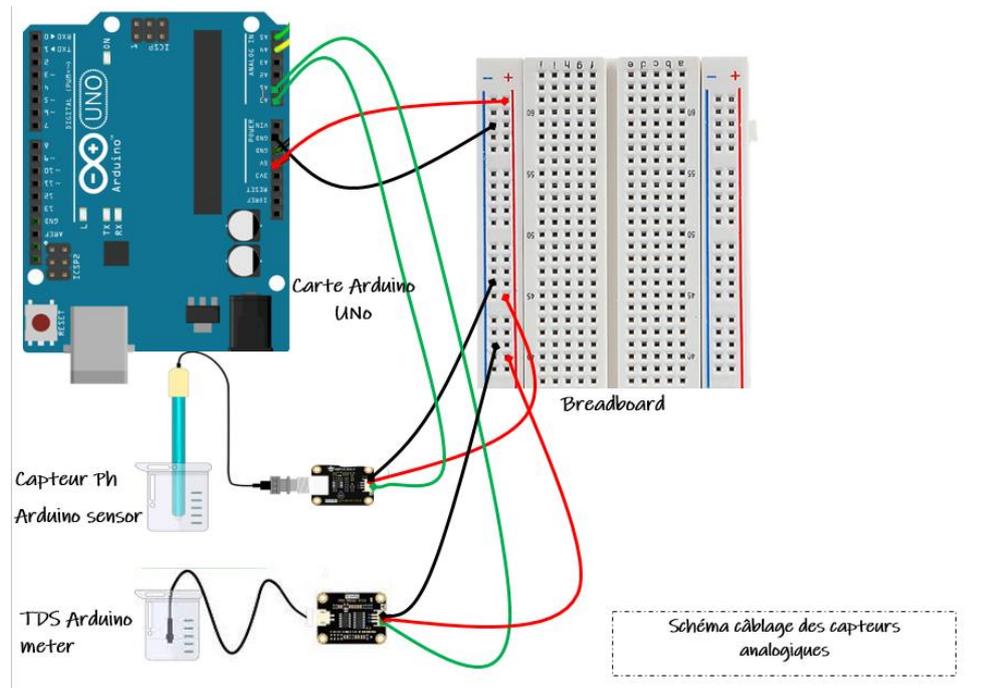
def destroy():
    GPIO.output(RelayPin, GPIO.HIGH)
    GPIO.cleanup()

if __name__ == '__main__': # début du programme
    setup()
    try:
        loop()
    except KeyboardInterrupt: # Arrêt 'Ctrl+C'
        destroy()
```

3 Capteur analogique

3.1 Test Capteur de pH avec connecteur BNC et capteur TDS

Branchement



Code test arduino

Ce code va afficher dans le moniteur toutes les secondes la valeur du ph et de l'uc.

```
#define Offsetph 0.00 // étalonnage ph
#define Offsettuc 0.00 // étalonnage uc
float tdsvalue = 0;
void setup()
{
  Serial.begin(9600);
}
void loop() {
  int sensorValue = analogRead(A0);
  int SensorValuetds= analogRead(A1);
  float voltage = sensorValue * (5.0/1024.0);
  float voltagetds = SensorValuetds * (5.0/1024.0);
  //Serial.println(voltage);
  float phvalue = (3.5*voltage)+Offsetph;
  float tdsvalue = ((133.42*voltagetds*voltagetds*voltagetds - 255.86*voltagetds*voltagetds + 857.39*voltagetds)*0.5)+ Offsettuc;
  Serial.print(phvalue);
  Serial.print(" ");
  Serial.println(tdsvalue);
  delay(1000);
}
```

Code test python

L'objectif est de récupérer les valeurs affichées dans le moniteur

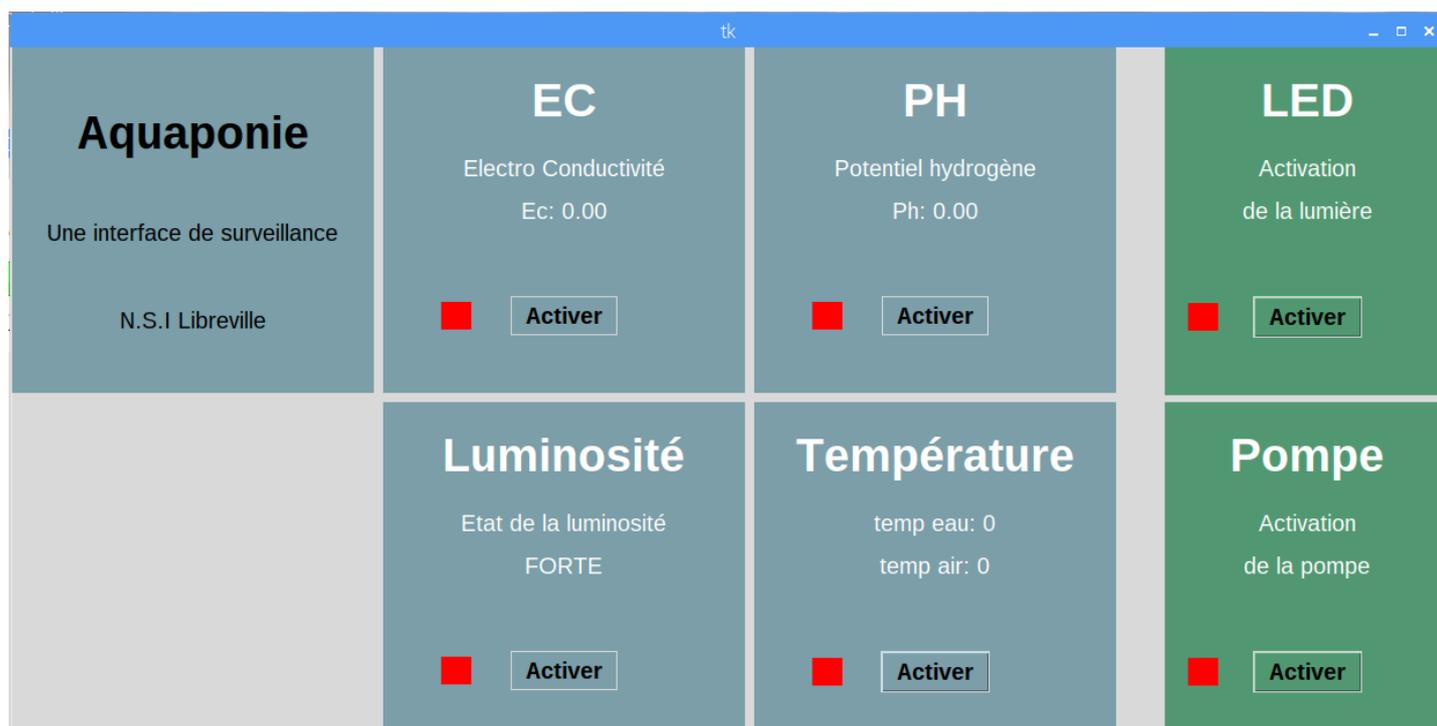
```
# Import de la librairie serial
from serial import Serial

# Ouverture du port serie avec :
# '/dev/ttyXXXX' : definition du port d ecoute (remplacer 'X' par le bon nom)
# 9600 : vitesse de communication
serialArduino = serial.Serial('/dev/ttyACM0', 9600)

#Ecriture de chaque message recu
while True :
    print(serialArduino.readline())
```

4 Le Projet Final Tkinter

L'objectif est de proposer une version de surveillance du système aquaponique sans utilisation de serveurs. Le principe est basé sur une interface Tkinter



CAPTURE ECRAN DU TKINTER. LES BOUTONS PERMETTENT DE PRENDRE LES MESURES OU D'AGIR SUR LES ACTIONNEURS

Le code complet est disponible en téléchargement

5 MariaDb / MySQL

Source : <https://ressourcesinformatiques.com/article.php?article=31f>

5.1 Installation MariaDb _MySQL

Mise à jour de la Raspberry Pi et installation de MariaDb

```
sudo apt-get update  
sudo apt-get install mariadb-server mariadb-client
```

Vérification de l'installation de MariaDb

```
ps -e | grep mysql
```

Connexion au serveur SQL et création d'un utilisateur. Ici l'utilisateur se nomme « pi » avec mdp « raspberry ».

```
sudo mysql -u root  
CREATE USER 'pi'@'localhost' IDENTIFIED BY 'raspberry';
```

Accorder tous les droits à l'utilisateur 'pi'

```
GRANT ALL PRIVILEGES ON *.* TO 'pi'@'localhost';
```

Création database aquaponie

```
CREATE DATABASE aquaponie;  
USE aquaponie;
```

Vérification de l'installation de la database

Connexion au serveur avec root

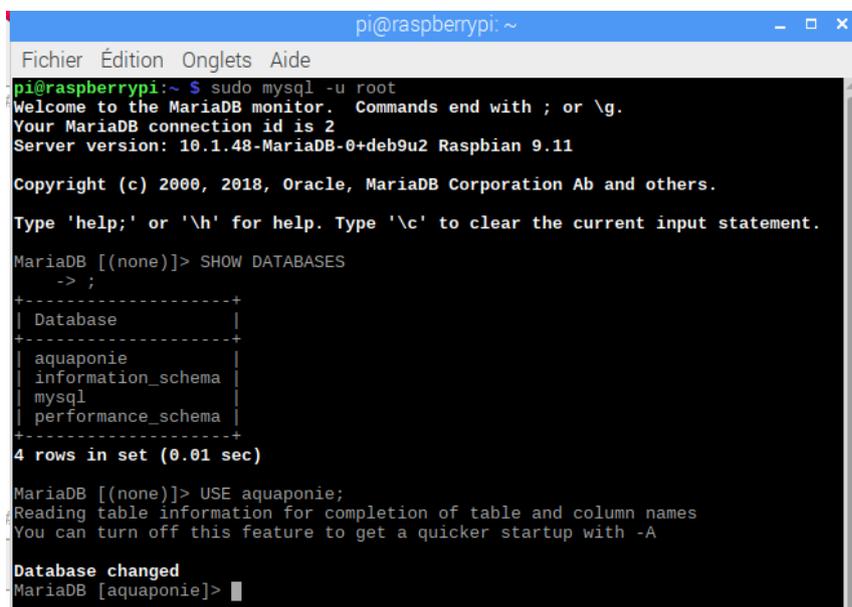
```
sudo mysql -u root
```

Affichage des bases

```
SHOW DATABASES ;
```

Utilisation de la table aquaponie

```
USE aquaponie ;
```



5.2 Gestion de la base de données depuis Python _ phase de test

Préambule

- Gestion de la base de données depuis Python, la bibliothèque mysql.connector doit être installée

Avant d'installer, il est important d'exécuter update et upgrade ;

```
sudo apt-get update
sudo apt-get upgrade
```

Installation

```
sudo apt-get install python3-mysqldb et cela fonctionne sur pi-3
```

Code python test

Programme test. L'objectif est de vérifier la création d'une relation Mesures. Cette table n'est pas la définitive, il s'agit d'un test. Nous allons également affecter les valeurs (1, 30.24,25.45,0,mesure_date) correspondant à la variable mesure

Mesures	
<u>Id</u>	INT
temperature_eau	FLOAT
temperature_carte	FLOAT
indicateur_luminosite	INT
date_heure	DATETIME

```

import mysql.connector
from datetime import datetime

#affectation de la variable mesure_date
now = datetime.now()
mesure_date = now.strftime('%Y-%m-%d %H:%M:%S')

#Se connecter à la database avec l'utilisateur et son mdp
conn = mysql.connector.connect(host="127.0.0.1",
                               user="pi", password="raspberrypi",
                               database="aquaponie")
cursor = conn.cursor()

# Création de la table
cursor.execute("""
CREATE TABLE IF NOT EXISTS Mesures (
id INT NOT NULL PRIMARY KEY,
temperature_eau FLOAT,
temperature_carte FLOAT,
indicateur_luminosite INT,
date_heure DATETIME)
; """)

#insertion des valeurs dans la relation
mesure = (1, 30.24,25.45,0,mesure_date)
cursor.execute("""INSERT INTO Mesures
(id,temperature_eau,temperature_carte,indicateur_luminosite,
date_heure) VALUES(%s, %s, %s, %s, %s)""", mesure)

#Affectation et déconnexion
conn.commit()
conn.close()

```

Vérification

Connection au serveur avec root, Utilisation de la table aquaponie

```

sudo mysql -u root
USE aquaponie ;
SELECT * FROM Mesuresquit ;

```

```

pi@raspberrypi: ~
Fichier  Édition  Onglets  Aide
pi@raspberrypi:~$ sudo mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 6
Server version: 10.1.48-MariaDB-0+deb9u2 Raspbian 9.11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> USE aquaponie;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [aquaponie]> SELECT * FROM Mesures;
+----+-----+-----+-----+-----+
| id | temperature_eau | temperature_carte | indicateur_luminosite | date_heure |
+----+-----+-----+-----+-----+
| 1  | 30.24          | 25.45             | 0                     | 2023-04-19 20:47:28 |
+----+-----+-----+-----+-----+
1 row in set (0.01 sec)

MariaDB [aquaponie]>

```

Effacer la table test

```
DROP TABLE Mesures ;
```

Quitter mysql

```
quit
```

6 Installation du serveur Apache

Tout d'abord, nous allons installer Apache, qui est **le serveur web en tant que tel**.

Mise à jour

```
sudo apt update
```

Mise à niveau des packages installés

```
sudo apt upgrade
```

Installation du serveur Apache.

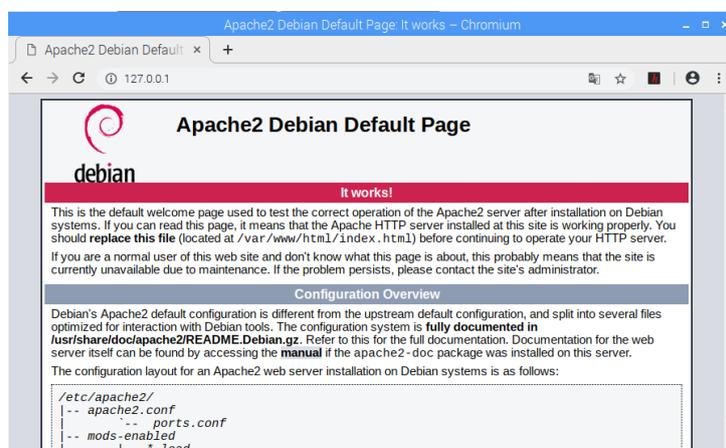
```
sudo apt install apache2
```

Réglage droits du dossier d'apache pour facilement administrer les sites. Pour cela, lancez les commandes suivantes :

```
sudo chown -R pi:www-data /var/www/html/  
sudo chmod -R 770 /var/www/html/
```

Vérifier qu'Apache fonctionne

Ouvrir le navigateur web de la Raspberry et aller à l'adresse «<http://127.0.0.1>». Vous devriez alors obtenir cette page



Apache utilise le répertoire `/var/www/html` comme racine pour votre site. Cela signifie que quand vous appelez votre Raspberry sur le port 80 (http), Apache cherche le fichier dans `/var/www/html`. Par exemple, si vous appelez l'adresse `http://127.0.0.1/example`, Apache cherchera le fichier `example` dans le répertoire `/var/www/html`.

Pour ajouter de nouveaux fichiers, sites, etc., vous devrez donc les ajouter dans ce répertoire.

7 Installation de PHP sur la Raspberry

Nous allons installer l'interpréteur, afin d'utiliser le langage PHP. Le langage PHP est principalement utilisé pour rendre un site dynamique.

Installation

```
sudo apt install php php-mbstring
```

#Vérifier que PHP fonctionne

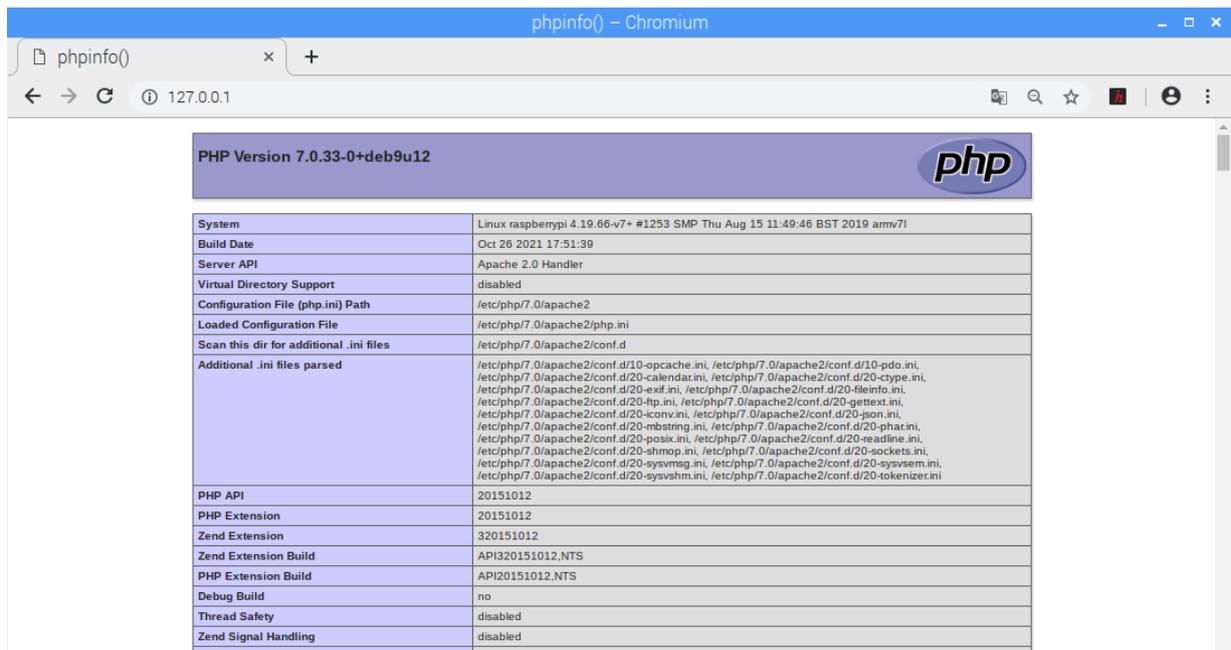
Pour savoir si PHP fonctionne correctement, la méthode est relativement proche de celle employée pour Apache. Vous allez en premier lieu supprimer le fichier « index.html » dans le répertoire « /var/www/html ».

```
sudo rm /var/www/html/index.html
```

Créer un fichier « index.php » dans ce répertoire

```
echo "<?php phpinfo(); ?>" > /var/www/html/index.php
```

Ouvrir le navigateur web de la Raspberry et aller à l'adresse «http://127.0.0.1». Vous devriez alors obtenir cette page



The screenshot shows a Chromium browser window displaying the output of the PHP info command. The page title is "phpinfo() - Chromium" and the address bar shows "127.0.0.1". The main content is a table with the following information:

PHP Version 7.0.33-0+deb9u12	
System	Linux raspberrypi 4.19.66-v7+ #1253 SMP Thu Aug 15 11:49:46 BST 2019 armv7l
Build Date	Oct 26 2021 17:51:39
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-mbstring.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini, /etc/php/7.0/apache2/conf.d/20-sockets.ini, /etc/php/7.0/apache2/conf.d/20-sysmsg.ini, /etc/php/7.0/apache2/conf.d/20-syssem.ini, /etc/php/7.0/apache2/conf.d/20-sysvshm.ini, /etc/php/7.0/apache2/conf.d/20-tokenizer.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012.NTS
PHP Extension Build	API20151012.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	disabled

8 Installation de PHPMYadmin

PHPMyAdmin est une application développée en PHP, et qui vise à fournir une interface simplifiée pour MySQL. L'installation de PHPMYAdmin n'est pas du tout obligatoire. Nous ferons ici une installation sans paramètres de sécurité particuliers.

#Installation

```
sudo apt install phpmyadmin
```

PHPMyAdmin vous posera plusieurs questions concernant ses réglages.

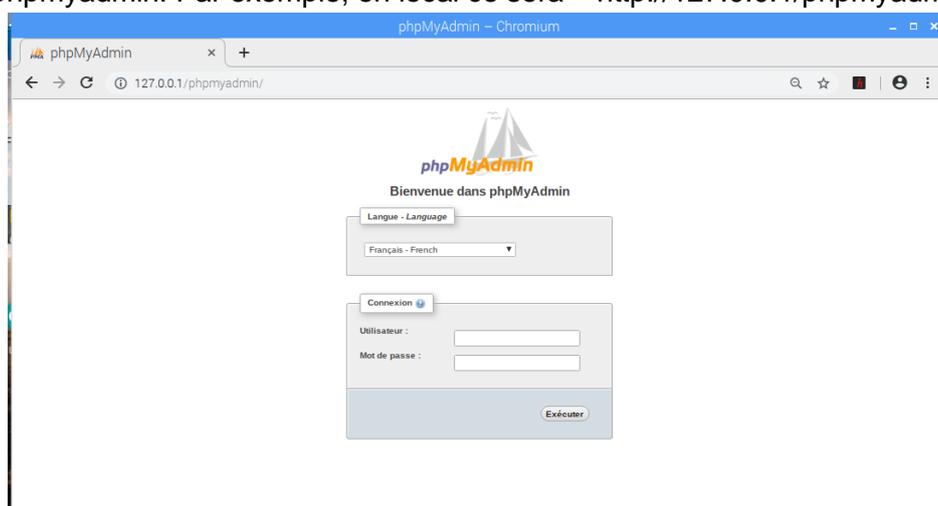
- Choisissez d'utiliser PHPMYAdmin pour un serveur Apache2.
- Comme nous avons déjà configuré mysql, choisissez **non** à la question concernant l'utilisation de dbconfig-common.

activer l'extension mysqli

```
sudo phpenmod mysqli  
sudo /etc/init.d/apache2 restart
```

#Vérifier l'installation de PHPMYAdmin

Pour vérifier le bon fonctionnement de PHPMYAdmin, utiliser l'adresse de votre Raspberry suivi de /phpmyadmin. Par exemple, en local ce sera « <http://127.0.0.1/phpmyadmin> ».



Si jamais, vous avez une erreur, cela peut venir du fait que PHPMYAdmin se soit installé dans un autre dossier. Dans ce cas, essayez la commande

```
sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin
```

9 Projet final

9.1 Le fichier python

Le fichier python **aquaponie.py** à lancer pour prendre les mesures. Pour rappel l'étape de planification de tâche n'a pas abouti. L'idéal aurait été de prendre par exemple les mesures et la photo 2 fois par jour. Le fichier python comprend :

les fonctions de prises de mesures :

- def prendre (ph) qui retourne la valeur du ph (prise sur le moniteur arduino)
- def prendre-ec() qui retourne la valeur EC (prise sur le moniteur arduino)
- def temperature() qui retourne une list de 2 float corresspondant à la température des 2 sondes (l'eau et de l'air)
- def camera() qui prend la photo et la stocke dans le répertoire. il est possible de nommer la photo par la date et heure de prise de vue.
- def luminosite () qui retourne l'indice de luminosité (0 ou 1)

la création de la table « Mesure » si elle n'existe pas

```
cursor.execute("""
CREATE TABLE IF NOT EXISTS Mesure(
date_heure DATETIME ,
temperature_eau FLOAT,
temperature_carte FLOAT,
indicateur_luminosite INT,
Ec FLOAT,
Ph FLOAT)
; """)
```

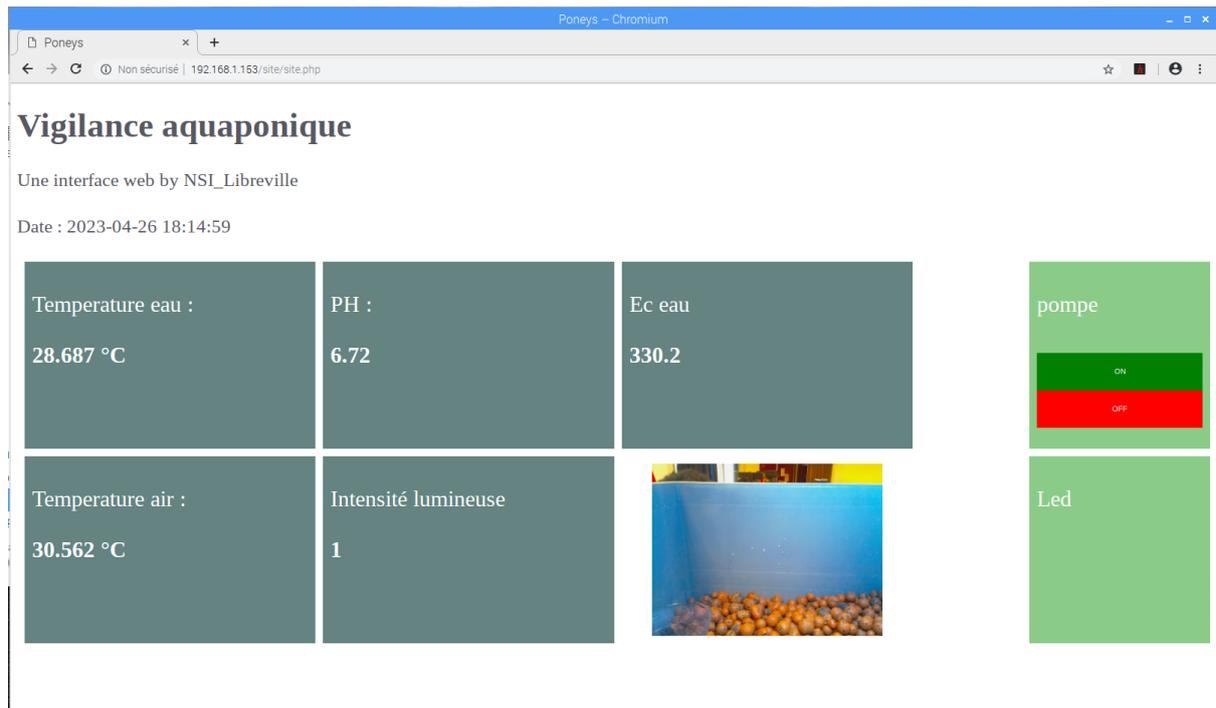
Les prises de mesures en utilisant les fonctions ci-dessus

```
camera()
t_eau=temperature()[0]
t_carte=temperature()[1]
luminosite=luminosite()
now = datetime.now()
n_ec = prendre_ec()
n_ph = prendre_ph()
date= now.strftime('%Y-%m-%d %H:%M:%S')
```

Insertion dans la base de données des mesures

```
mesure = (date,t_eau,t_carte,luminosite,n_ec,n_ph)
cursor.execute("""INSERT INTO Mesure
(date_heure,temperature_eau,temperature_carte,indicateur_luminosite,Ec,Ph) VALUES(%s, %s, %s, %s, %s, %s)""", mesure)
```

9.2 Le fichier php et css



CAPTURE D'ECRAN DE LA PAGE ADRESSE/SITE/SITE.PHP

9.3 Code php pour affecter des variables

- Nous indiquons le nom de l'utilisateur, son mdp, la base de données

```
define('DB_HOST' , 'localhost');  
define('DB_NAME' , 'aquaponie');  
define('DB_USER' , "pi");  
define('DB_PASS' , "raspberry");
```

- Commande sql pour rechercher la dernière entrée

```
// sélection de la dernière entrée de la table temphumi triée par date.  
$sql = 'SELECT * FROM Mesure ORDER BY date_heure DESC LIMIT 1 ';
```

- On affecte les variables correspondant aux attributs de la ligne

```
// on remplit nos variables avec les datas  
while ($valeur = $reponse->fetch())  
{  
    $date_heure = $valeur['date_heure'];  
    $temperature_eau = $valeur['temperature_eau'];  
    $temperature_carte = $valeur['temperature_carte'];  
    $indicateur_luminosite = $valeur['indicateur_luminosite'];  
    $Ec = $valeur['Ec'];  
    $Ph = $valeur['Ph'];  
}
```

9.4 Code html

Le code html est classique avec un système de boîte <div>. Dans chacune des boîtes, nous pouvons afficher la variable créée plus haut avec du code php.

Exemple pour le ph , le code va afficher PH en titre 3 avec sa valeur après un retour chariot

```
<?php
echo "<h3>PH :<br>
<br>
<strong>$Ph</strong> </h3>";
?>
```

Le code complet est disponible en téléchargement