**Solutions et outils utilisés**

**a) Base de données MariaDB**

* **MariaDB** est une base de données relationnelle open-source largement utilisée, dérivée de MySQL.
* **Outils associés** :
  + **phpMyAdmin** : Interface web pour gérer la base de données MariaDB.
  + **MySQL Workbench** : Outil de conception de base de données et gestion de SQL.
* **Avantages** : Performance, sécurité, et possibilité de s'intégrer facilement avec PHP.

En ligne de commande sur linux avec l’outil de commande « apt »

**b) Serveur applicatif PHP**

* **PHP** : Langage de programmation côté serveur, largement utilisé pour développer des applications web.
* **Serveur Apache** : Serveur HTTP open-source qui héberge des applications PHP.
* **MQTT** : Protocole léger de messagerie, utilisé ici pour transmettre des données depuis les capteurs. Pour intégrer MQTT à PHP, des bibliothèques comme **phpMQTT** ou **Mosquitto PHP extension** peuvent être utilisées.
* **IDE pour PHP** :
  + **Visual Studio Code** : IDE léger et populaire avec une riche extension pour PHP.
* **Avantages** : PHP est largement utilisé pour les serveurs web, avec un large écosystème de bibliothèques.

**c) Alertes par email**

* Pour l'envoi d'emails, **PHPMailer** ou **SwiftMailer** peuvent être utilisés dans le code PHP. Ces bibliothèques permettent d'envoyer des emails avec des fonctionnalités avancées comme le support de pièces jointes, l'authentification SMTP, etc.
* **Configuration SMTP** : Pour les emails d'alerte, des services comme **Mailgun**, **SendGrid** ou l'utilisation d'un serveur SMTP local peuvent être envisagés.

**d) Environnement Linux**

* **Système d'exploitation** : Une distribution Linux, **Debian** (Debian12, que je connais et que j’ai eu l’habitude d’utiliser pour les différents TP en cours), qui offre un environnement stable pour installer Apache, MariaDB, et PHP.qpt

**2. Test et comparatif des solutions**

**a) Base de données : MariaDB**

* **MariaDB** : Léger, bonne performance pour des applications web standards.

**b) Serveur web et PHP**

* **Apache**: Gratuit, performance suffisante pour notre projet et plus simple d’utilisation que d’autre

**c) MQTT**

* **Mosquitto** : Le serveur MQTT le plus populaire, facile à installer et à configurer.

**d) Éditeur/IDE**

* **Visual Studio Code** : Gratuit, léger et rapide, avec de nombreuses extensions pour PHP.

**e) Passerelle**

* Les passerelles LAIRD ou des alternatives comme **Raspberry Pi** avec un module de communication sans fil sont suffisantes pour des projets à petite ou moyenne échelle.
* **Raspberry Pi** peut également être utilisé pour une gestion locale des capteurs et de la communication MQTT.

Logiciel schéma MCD et MLD :

* Looping (utilisé)
* JMerise

**3. Devis estimatif et budget**

Voici une estimation des coûts pour un environnement de développement typique :

**a) Matériel**

* **Raspberry Pi 4** : ???

**b) Logiciels**

* **MariaDB** : Gratuit (open-source)
* **Apache** : Gratuit (open-source)
* **PHP** : Gratuit (open-source)
* **Mosquitto (serveur MQTT)** : Gratuit (open-source)

**d) Coût total estimé :**

* **Matériel de base** (1 Raspberry Pi + capteurs) : ???
* **Coût logiciel et services** : Gratuit

**Conclusion :**

* **Choix des outils** : La combinaison de **MariaDB**, **PHP**, **Apache**, **Mosquitto** (MQTT) et **Raspberry Pi** est une solution robuste et adaptée à ce projet.
* **Budget** : Un budget de ???pourrait être nécessaire pour couvrir le matériel

Lien utile installation :

<https://www.it-connect.fr/installer-un-serveur-lamp-linux-apache-mariadb-php-sous-debian-11/>

https://www.phpmyadmin.net/

<https://www.jfreesoft.com/JMerise/>

https://www.aranacorp.com/fr/mettre-en-place-un-serveur-mqtt-avec-mosquitto/