

Plan de présentation :

- Présentation du sujet de TFE

- Titre : Plateforme web visant à afficher les mesures prises par une Raspberry Pi avec différents capteurs environnementaux et de gaz.
- Quoi :
Plateforme web reliée à un appareil de prise de mesures qui récupère différentes valeurs.
Cet appareil est proposé à la carte (base générique) : température, humidité, pression atmosphérique, eCO2, COVT (= qualité de l'air).
- Problématique :
 - Récolter des informations afin d'optimiser l'environnement du personnel et améliorer l'état de santé ainsi que la productivité de l'être humain,
 - Alerter en cas d'un éventuel danger dans un milieu de travail à risque.
- Comment :
J'ai eu cette idée peu avant le début de l'année académique en cours et j'ai soumis cette idée à une entreprise, Dreamnet SPRL, qui y voit un certain intérêt.
Le professeur A. Dewulf a également un client potentiellement intéressé et qui aurait besoin d'un capteur détectant l'Oxyde de Silicium.

- Méthodologie et analyse

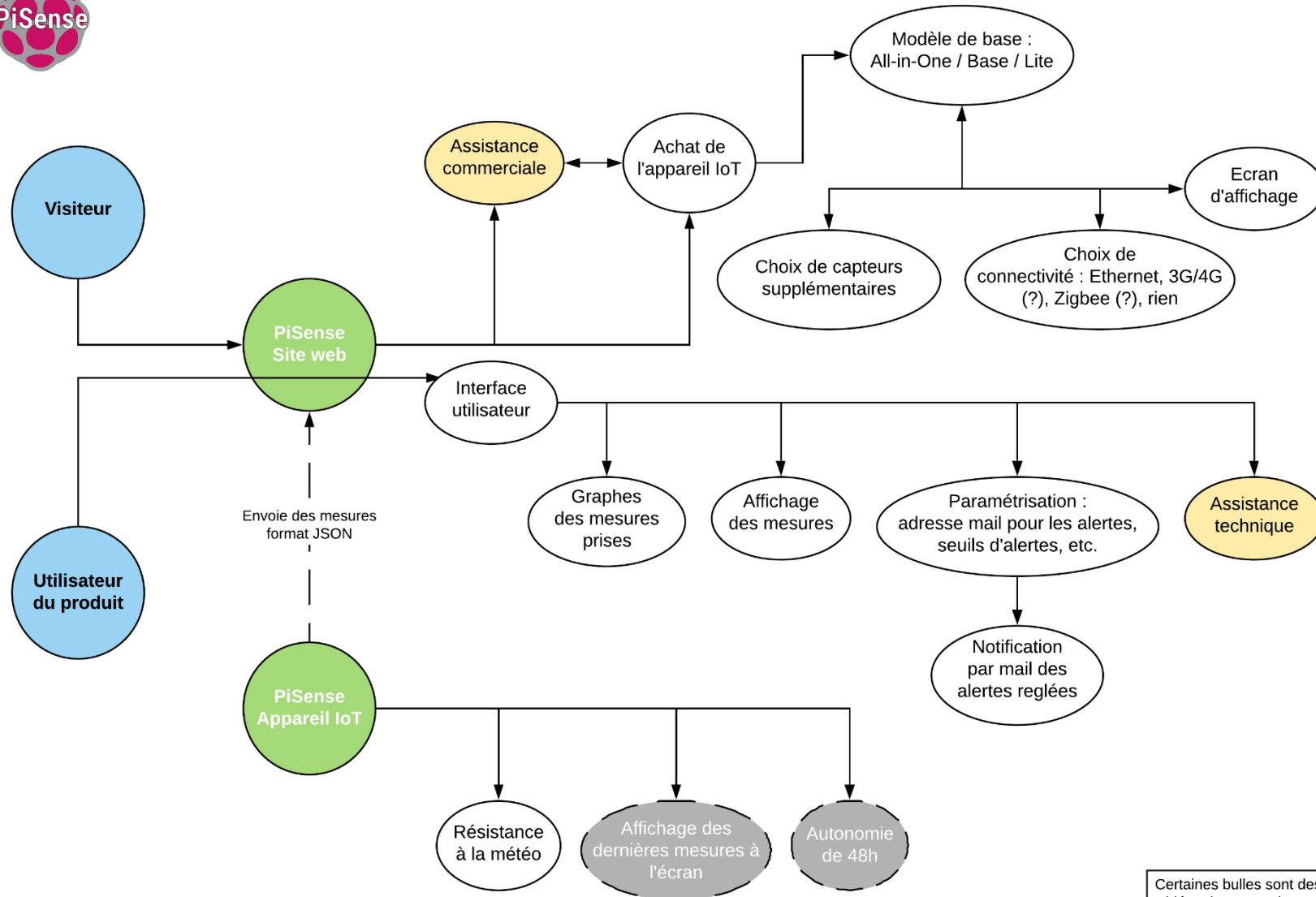
- Méthodologie :
Utilisation de l'outil de versioning Git hébergé sur GitHub.
Utilisation de cartes et planning pour mieux cibler le développement.
Utilisation de l'environnement Docker pour le déploiement des services web.
- Technologies :
 - Site web : HTML/CSS/JS + librairie React
 - Serveur web : NGINX
 - IoT (Raspberry + les capteurs) : Python
 - API : type REST ? (échange de données en format JSON)
 - DB
- Analyse :
Aucun capteur disponible ne mesurant uniquement que l'Oxyde de Silicium, que des capteurs analysant la qualité de l'air par le biais des fines particules (PM2.5/PM10 – PMx désigne les particules dont le diamètre est inférieur à x micromètres).
L'API est nécessaire parce que React ne peut pas se connecter à une base de données directement. L'API interagit avec la DB tout comme JS (pour demander les mesures à afficher sur le site). L'IoT interagit avec l'API pour envoyer les mesures à stocker dans la DB.
NGINX choisi comme webserver pour sa rapidité et l'intérêt derrière sa configuration détaillée et le Reverse Proxy (permet la relation entre plusieurs serveurs).
Commencement du développement de la partie IoT avant d'entamer la partie services web en parallèle et terminer par le site web.

- État d'avancement (voir page 3 pour plus de détails)

- Analyse technique réalisée afin de mieux visualiser la plupart des contraintes à rencontrer,
- 3 des 4 capteurs reçus et soudés (+ testé), attente du capteur SiO,
- Modèle de ce à quoi ressemblerait potentiellement l'accueil du site réalisé,
- Code Python d'un capteur déjà entamé.



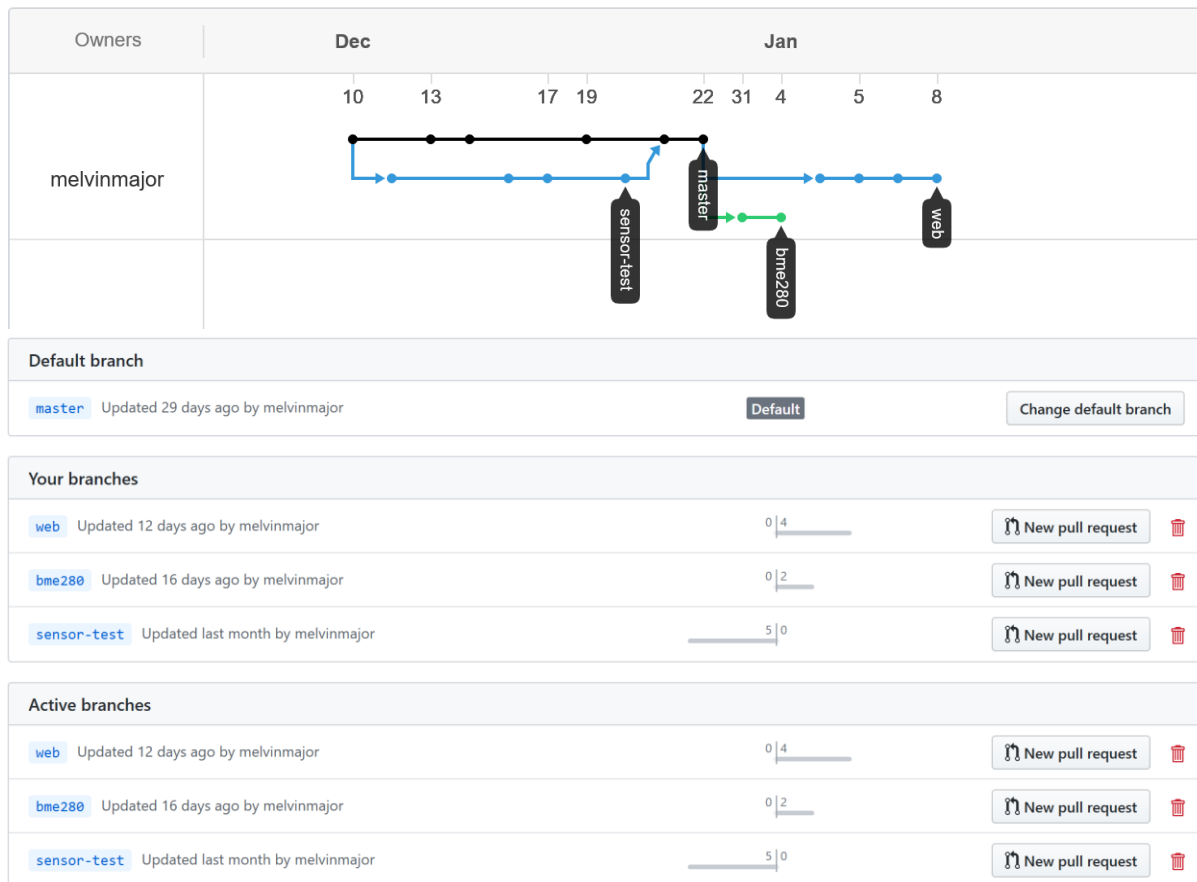
PiSense - Schéma d'analyse
Melvin Campos Casares



Certaines bulles sont des idées de conception non obligatoires (généralement affichées en bulles grises + contour en tiret).

Repository GitHub : <https://github.com/melvinmajor/tfe-PiSense>

État d'avancement :



- ☒ Recherche des capteurs intéressants pour le sujet de TFE
- ☒ Cahier de charges
- ☒ Analyse technique
- ☒ Apprentissage de Python : cours sur OpenClassrooms
- ☒ Recherche des capteurs intéressants pour le sujet de TFE
- ☒ Réception et soudures sur les premiers capteurs retenus (3/4 achevé)
- ☒ Début du développement du code Python des capteurs
- ☒ Maquette de la page d'accueil du site internet
- ☐ Apprentissage de React.JS : cours sur OpenClassrooms
- ☐ Apprentissage de REST : cours sur OpenClassrooms
- ☐ Commande OVH : VPS et nom de domaine
- ☐ Configuration du VPS et du DNS
- ☐ Mise en place du webserver NGINX
- ☐ Début du développement de l'API et de la base de données
- ☐ Intégration de diverses méthodes de connexion de l'appareil IoT : 3G/4G, etc.
- ☒ Début du développement du site web
(pour l'instant que la page d'accueil en HTML/CSS afin de correspondre à la maquette)
- ☐ Site web responsive
- ☐ Phase de test dans divers environnements