AQROPOL

Rapport

Participants au projet :

Jean-Baptiste Bordier
Manon Derocles
Mélian Ferrachat
Julien Garnier
Alan Gaubert
Adrien Leblanc
Thibault Leclair
Antoine Leval
Pierre Miola
Bashar Nemeh
Noureddine Kadri
Antoine Posnic
Fanny Prieur
Hantavolaniaina Sylvia Rabe

Mr François Bodin
Enseignant référent du projet AQROPOL
Enseignant chercheur à l'université de Rennes 1(IRISA)

Table des matières

I. Introduction

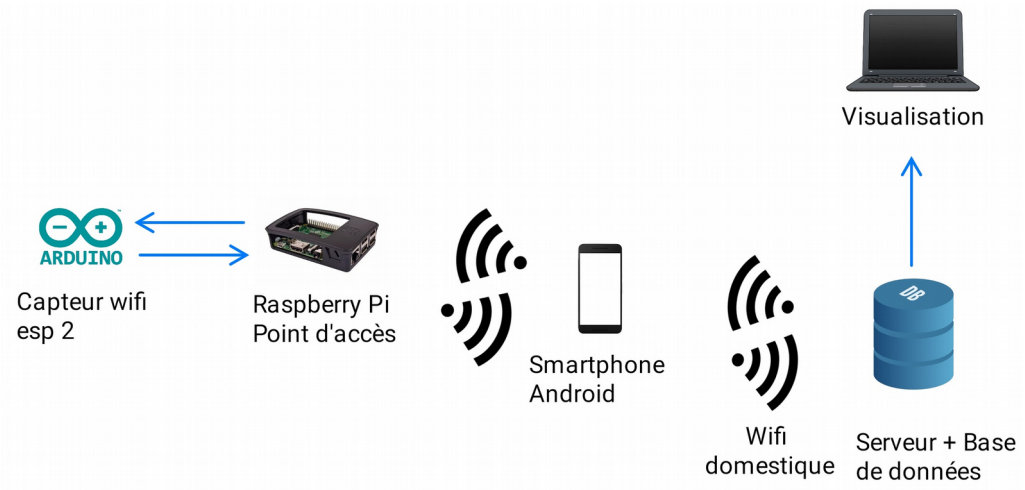
Ce rapport constitue l'aboutissement du travail accompli par notre équipe au sein du projet AQROPOL, de septembre 2017 à mai 2018, encadré par M. Bodin. Il présente d'abord une vision d'ensemble du projet, de la problématique associée et de la méthodologie de travail que nous avons appliquée et enfin les résultats obtenus.

A. Présentation du projet

L'objectif du projet AQROPOL est la création d'un système de récolte de données modulable, permettant de supporter le rassemblement d'informations de tout type sur une grande zone géographique, ici la ville de Rennes.

Il doit par exemple permettre d'obtenir des informations sur la qualité de l'air en rassemblant les données mesurées par des capteurs de pollution placés sur les bus du réseau de transports en commun. Une autre application possible est la possibilité de collecter des données à propos de la fréquentation réelle des amphithéâtres des universités. Cela permettrait de prévoir les pics de fréquentation des transports en commun et ainsi de pouvoir réagir en conséquence afin d'éviter une saturation ou une sous fréquentation du réseau, que ce soit au début ou à la fin des cours.

La figure *Fig.1* représente l'ensemble du projet AQROPOL, nous observons en effet notre capteur de pollution de l'air greffé sur un Raspberry Pi celui-ci communiquant en Wifi avec un smartphone Android, communiquant avec la base de données qui elle, envoie les



données à la visualisation et dans ce sens uniquement.

Fig.1: Schéma du fonctionnement du projet AQROPOL

Les applications sont nombreuses et ne demandent qu'à être trouvées. Pour notre part, nous avons dirigé notre réflexion vers la récolte de données concernant la pollution de l'air à Rennes.

B. Problématique

Le projet AQROPOL vise à réduire le coût budgétaire qu'impose le déploiement de projet de ce genre à grande échelle.

Il propose un moyen de rassemblement des données se reposant sur la coopération de smartphones de particuliers bénévoles. Ces particuliers collectent les données stockées au niveau des capteurs, avant de les envoyer au serveur qui se chargera de les centraliser sur une même base de données, dès qu'ils disposent d'une connexion internet.

De cette manière, le projet AQROPOL évite des frais supplémentaires liés à d'éventuels abonnements Internet et autres pour la récupération des données des capteurs, qui pourraient dissuader la mise en place d'un tel système.

C. Approche du problème dans le contexte

Comme précisé précédemment, nous avons poussé notre réflexion sur la récupération des données de pollution dans Rennes. Notre système devait donc être capable de remonter les données sur une large couverture géographique.

Pour rappel, notre système est modulaire. Ce choix vient d'un entretien avec notre client, qui après avoir situé le contexte du projet a voulu le rendre plus large afin de permettre d'implémenter de nouveaux capteurs ayant une fonction différente de ceux installés initialement, tels qu'une caméra infrarouge ou un sniffeur de réseaux Wifi. L'équipe a souhaité conserver le contexte initial de la pollution pour la guider. Dans ce contexte, les capteurs étaient répartis dans différentes zones et dispatchés de manière aléatoire permettant de couvrir entièrement la ville de Rennes. Prenant compte de l'absence de main d'oeuvre pour assurer la maintenance par souci d'économie et de logistique, nous avons introduit un nouveau type d'acteur bénévole, le smartphone. En effet, la présence de clef 3G/4G sur chaque capteur aurait fait grimper le budget à cause des abonnements. C'est pourquoi les usagers possédant un smartphone sont le nerf de notre économie de budget. Ils assurent le

relais de nos données via une application Android que l'équipe a développé.

Cette approche avec un intermédiaire est l'une des réponses possibles à la problématique et nous permet donc d'avoir la certitude que dans une zone géographique comme Rennes, des personnes utilisant des smartphones passeront à côté de nos capteurs.

D. Plan du Rapport

Nous allons maintenant vous présenter l'organisation du rapport pour guider le lecteur dans les explications du projet et montrer le lien ainsi que le raisonnement qu'il y a derrière.

Tout le long du projet, l'équipe a été découpée en plusieurs groupes afin de gérer le plus efficacement les tâches de celui-ci. Il y avait donc un groupe Capteur, Android, Serveur et Visualisation. Ce découpage se retranscrit dans chaque partie où chaque groupe aborde son point de vue et transmet ses expériences. Nous abordons en premier le processus du développement pour expliquer notre organisation ainsi que les solutions choisies par groupe. Après quoi, il nous a semblé important de revenir sur les décisions prises et les solutions adoptées dans le projet, d'analyser leurs points forts, leurs limites dans la partie Dimension d'analyse et critique. Enfin, on conclut le document par une synthèse de toutes les parties reprenant les points importants.

Cette organisation se calque sur celle d'un projet, où l'on commence d'abord par éclaircir le contexte et la problématique pour ensuite fournir une solution tout en la justifiant d'un point de vue critique.

II. Processus de développement

Le projet a connu plusieures phases avant d'aboutir au résultat présent. Nous nous sommes d'abord organisé par groupe avant de fournir une solution adaptée pour chaque partie du projet. Nous avons ensuite établi un plan d'action permettant à toutes les parties du projet de communiquer entre elles et enfin nous avons établi des tests, pour valider la totalité du projet.

A. Découpage du projet

Notre système au sein du projet est composé tout d'abord de capteurs liés à un mini-ordinateur que l'on va nommer NUC. Ensuite, il y a le serveur qui centralise les données dans sa base de données. Il y a également le smartphone qui sert d'intermédiaire entre le NUC et le serveur. Enfin le système propose une visualisation des données stockées dans la base. Le projet a donc été divisé en quatres groupes, voir *Fig.2*, chacun correspondant à un acteur dans notre système : Capteur, Android, Serveur et Visualisation.

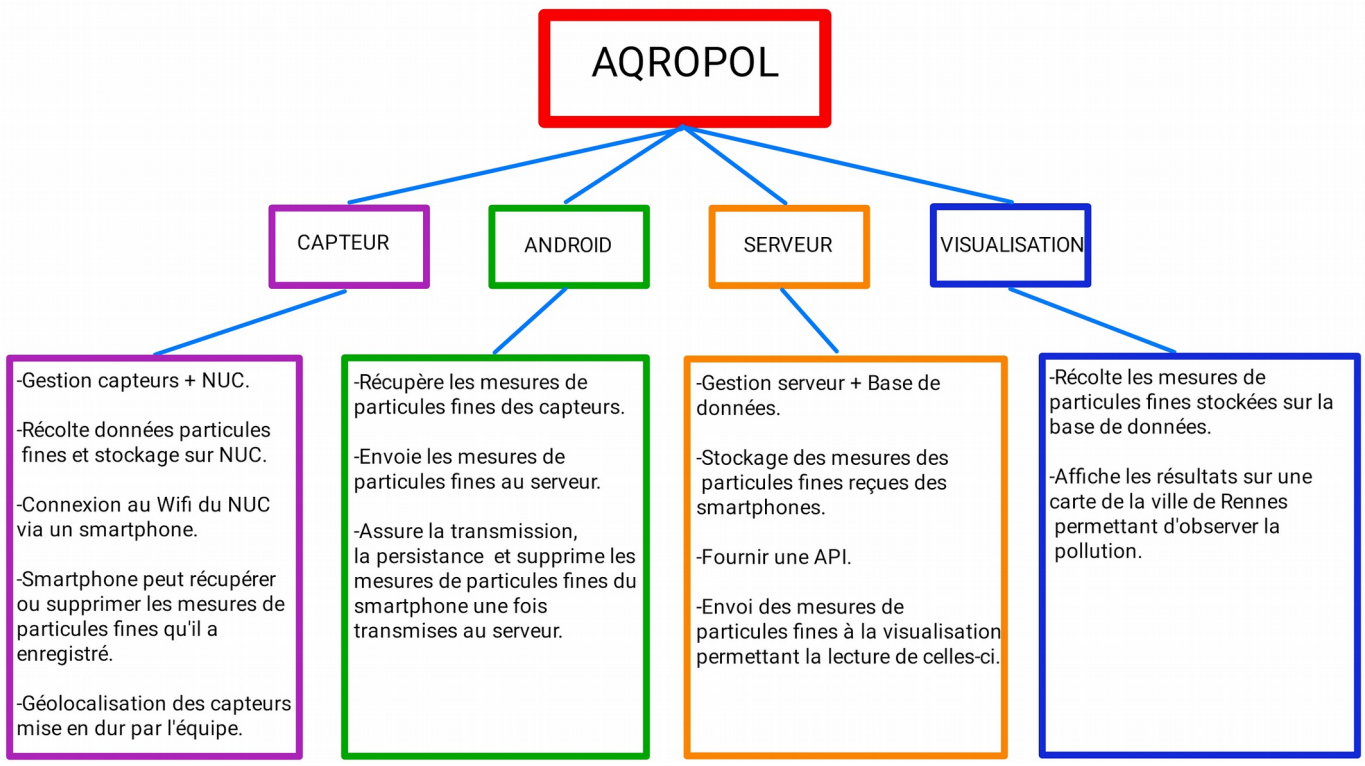


Fig 2: Schéma de répartition et rôle des groupes du projet AQROPOL

1. Le groupe Capteur comprend la gestion des capteurs et du NUC. Il permet la récupération des données récoltées par les capteurs, pour les stocker sur le NUC. Il permet aussi à un téléphone de se connecter à la wifi du NUC pour récupérer les données sauvegardées ou de demander la suppression de mesures déjà présentes dans la base de données.
2. Le groupe Android concerne l'application du smartphone. Il récupère les données des capteurs et les envoie au serveur. Il assure leur bonne transmission, leur persistance et

les supprime du smartphone une fois qu'elles sont transmises.

3. Le groupe Serveur gère le serveur mais aussi la base de données. Il a pour objectif la mise en place d'une base de données qui servira de stockage aux données reçues en amont qui présentent:
    - Nom du Hub et date d'envoi.
    - Nom, type du Capteur.
    - Unité, valeur de la Mesure.
    - Date d'enregistrement, coordonnées gps, hash.

Le groupe Serveur fournit également une API pour:
- L'insertion des données reçues: Enregistrer les informations ci-dessus.
- La lecture des informations stockées: Interroger la base de données selon certains filtres tels que la date d'une mesure, sa valeur, son type, etc, pour ensuite les afficher.

4. Le groupe Visualisation s'applique, après avoir reçu les mesures stockées dans la base de données, à afficher les résultats sur un système de WebMapping de Rennes afin d'observer la pollution ambiante.

L'implication de ces groupes dans le projet et les solutions qu'ils ont proposé sont détaillées dans les deux sous parties qui suivent.

B. Fonctionnement des différentes parties du projet

Nous allons aborder les solutions adoptées par chaque groupe pour répondre aux problématiques auxquelles ils ont dû faire face afin d'assurer leur rôle dans notre système.

1. Solutions choisies par le groupe Capteur

Les données du/des capteurs sont récoltées via un système de routage AMQP. Chaque entité peut y publier des données au format JSON. Les données reçues par le consumer AMQP sont stockées dans une base de données relationnelle. Un serveur REST s'occupe ensuite de rendre accessible les données récoltées en utilisant notamment les annotations JAVA.

2. Solutions choisies par le groupe Android

Nous avons choisi de programmer notre application Android en utilisant la technologie HTTP REST pour gérer les communications avec le serveur et le NUC.

3. Solutions choisies par le groupe Serveur

Nous avons choisi d'utiliser le serveur Apache2 avec un serveur SGBD SQL, à savoir MySQL et nous avons réalisé une API en PHP/MySQL utilisant le format JSON.

4. Solutions choisies par le groupe Visualisation

Nous avons utilisé l'API de MapBox basée sur OpenStreetMap pour réaliser

l'intégralité de la visualisation. Nous l'avons complétée en ajoutant la possibilité de choisir un format de type "HeatMap", d'afficher les bâtiments en 3D et de trier les données selon une date ou un mois.

C. Tests de mise en oeuvre assurant le bon fonctionnement du projet

Tout un processus a été mis en oeuvre pour assurer la remise et le maintien du projet post-développement. Cela va des tests qui ont été organisés pour valider des fonctionnalités aux livrables écrits dans le but d'informer et d'aider toute personne susceptible de s'intéresser au projet.

L'organisation des tests a commencé d'abord par une mise à plat des différents tests qui existent et sur la participation de chaque groupe à ces tests. Cela a abouti en un document qui figure parmi les livrables, "Politique de tests", qui détaille chaque type de test, les groupes qui vont devoir les mettre en oeuvre, leur hiérarchie et les livrables à fournir. L'un d'eux est le dossier de tests qui contient l'intégralité des tests dits "Tests d'acceptation" et les formalise de façon à rapidement pouvoir déterminer si le projet passe le test ou non. Nous avons également rédigé des livrables comme le "Manuel utilisateur", la "Documentation développeur" et le "Dossier de spécifications". Chacun de ces livrables à un public ciblé et sa lecture est adaptée au niveau de compétence attendu.

La *Fig.3* ci-dessous représente le protocole des différents tests établis entre chaque partie/niveau du projet assurant ainsi le fonctionnement de celui-ci.

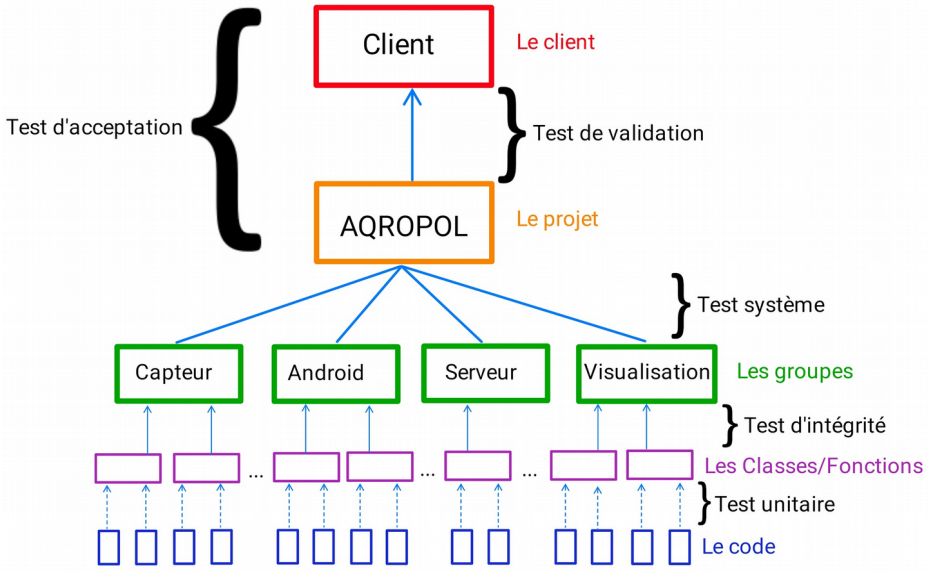


Fig 3: Schéma hiérarchique des tests établis

Grâce aux livrables fournis, il est plus simple pour une personne extérieure au projet de pouvoir s'impliquer dedans ou de le reprendre afin de le continuer ou l'adapter à ses besoins en ayant déjà une idée de ce qui a été fait en manière de test.

III. Dimension d’analyse / Critique

Les points forts du projet ainsi que ces limites vont être présentés dans cette partie. Ensuite nous exposerons les technologies utilisées pour la partie Capteur, Android, Serveur et pour la Visualisation.

A. Points forts du projet

Suite à nos choix et aux contraintes fixées par le client, nous avons distingué plusieurs points forts dans notre projet.

En effet il peut être implanté dans des zones non couvertes par internet ce qui à ce jour est très intéressant car notre système est capable de remonter les données par le biais d’intermédiaires, les smartphones, et d’attendre qu’ils aient internet pour récupérer leurs données. Il est très rare qu’un utilisateur de smartphone ne se connecte pas à Internet rien que pour tenir à jour son smartphone. Il faut également savoir que la visualisation des données reste très utile dans notre contexte pour sensibiliser les personnes à la pollution. N’oublions pas que le projet est adaptable à différent type de capteur et qu’il est facile d’utilisation pour toute sorte de personne. Son déploiement n’est pas contraint non plus par le coût qui est resté notre premier engagement.

Ces points forts viennent de décisions qui nous ont poussé à aller dans ce sens mais elles apportent également des limites car nous répondons à une problématique très précise.

B. Limites du projet

Ces limites peuvent être contraignantes, dès lors que nous sortons de notre contexte.

Notre système ne fournit pas une réponse en temps réel. Le temps d’arrivée des données jusqu'à la base de données varie selon les utilisateurs de smartphone. De ce fait, il est impossible de prédire combien de temps cela va prendre. Une autre faiblesse est que nous ne pouvons pas acheminer une grande quantité de données. Nous sommes limités par la mémoire des smartphones des utilisateurs car ils participent de manière bénévole. Nous ne pouvons nous permettre de réduire drastiquement leur capacité mémoire sans quoi ils risquent de ne plus participer au projet.

Ces limites proviennent de nos choix sur l’architecture du projet et du fait d’avoir choisi le smartphone comme intermédiaire entre les capteurs et le serveur.

C. Technologies utilisées

Plusieurs technologies ont été testé pour essayer de répondre à notre problème et

certaines ont été abandonnées pour d'autres moins contraignantes. Nous allons revenir sur ces technologies utilisées tout au long de l'implémentation du projet au sein des différents groupes.

1. Capteur

Nous avons développé tous nos modules en JAVA car nous étions plus à l'aise avec ce langage et nous n'avons eu aucun problème à trouver les librairies des différents protocoles utilisés.

La solution REST et Spring

Nous avons utilisé l'API REST basée sur HTTP et nous avons essayé de le rendre le plus standard possible avec le moins de code spécifique possible pour des questions d'évolution et de maintenance. C'est la raison aussi pour laquelle PHP a été écarté, avec notamment ses nombreuses mises à jour de sécurité. Nous avons également choisi Spring pour la partie web service. Ce framework offre une totale abstraction de toutes les parties récurrentes du code, implémente uniquement des standards, et est continuellement mise à jour sans action d'une personne tierce. On limite donc grandement le nombre de ligne de code de notre application et la reprise du logiciel et sa compréhension n'en sont que plus simples.

La solution RabbitMQ

Pour la communication entre le NUC et les capteurs, nous avons décidé d'utiliser une solution MQTT pour plusieurs raisons. Tout d'abord, cela permet l'ajout et la suppression d'un capteur à n'importe quel moment ; le capteur se connecte au broker et envoie un message sur un topic en particulier, chaque capteur ayant son topic. La récupération des données au niveau du NUC est elle aussi facilité, puisqu'il suffit d'écouter tous les topics disponibles à la racine du broker. Un protocole AMQP nous donnait aussi ces avantages, mais nous avons finalement choisi MQTT en utilisant un broker RabbitMQ car il ne nous demandait aucun temps d'apprentissage supplémentaire étant au programme du Master 1 Génie Logiciel au second semestre. La documentation de l'API JAVA est très fournie sur le site officiel qui contient aussi des exemples d'utilisations. RabbitMQ propose également une interface d'administration activable et très pratique durant la phase de développement ou pour monitorer les capteurs très simplement. De plus, ayant fait le choix d'utiliser SPRING pour la partie base de données et web service rabbitMQ étant implanté de façon standard en SPRING, son intégration s'est vue facilitée.

2. Android

Nous avons choisi de développer sous Android car l'interface se développe en XML et le reste de l'application en JAVA. L'équipe était plus à l'aise avec ces deux langages. Android est le système d'exploitation le plus utilisé dans le monde depuis 2011 (87,7% des téléphones mobile en 2017), c'est pour ce critère que nous avons choisi cet OS là. Nous avons utilisé l'API REST en accord avec le groupe Capteur et le groupe Serveur pour la communication des données pour les mêmes avantages énoncés plus haut.

3. Serveur

Pour la communication, suite à une délibération sur le sujet avec l'équipe Android, nous en sommes venus à implémenter les méthodes POST/GET de HTTP.

Serveur Apache, MySQL et Format JSON

Nous avons installé un serveur Apache2 sur le Raspberry Pi qui nous était confié afin d'ensuite pouvoir mettre en place la base de données à laquelle nous avions réfléchi. C'est ainsi que nous avons opté d'en implémenter une de type MySQL.

Les raisons principales pour lesquelles nous avons choisi ces technologies sont leurs simplicités d'installation, de configuration et d'utilisation qui correspondaient aux compétences de notre équipe. De plus, étant parmi les plus répandues, il est aisé de trouver de la documentation en ligne.

Afin de garantir un transfert de données réutilisable par tous les autres groupes du projet, nous nous sommes basés sur le format JSON. En effet, il permet de stocker des données de différents types et ne dépend d'aucun langage. Il est notamment pris en charge par de nombreux langages dont PHP, JavaScript, Java, etc.

4. Visualisation

Pour développer la partie visualisation des données récoltées nous avons décidé d'afficher les mesures sous forme de point sur une map. Ce format de visualisation nous était demandé dans le cahier des charges fourni par le client. Suite à cette demande, nous nous sommes renseignés sur plusieurs API d'affichage de map.

La solution MapBox

Nous nous sommes renseignés sur MapBox, une API basée elle même sur Open street map. Elle fonctionne selon les mêmes principes mais possède une documentation plus explicite, claire, organisée et un design pour l'affichage de nos données. Son implémentation nous semblait aussi beaucoup plus simple et plus proche de ce que nous connaissions déjà et un des membres l'avait déjà expérimenté. Nous avons donc commencé à implémenter notre page à l'aide de MapBox. La version de l'API utilisée est la 0.40.1 permettant entre autre d'afficher les bâtiments en 3D.

Format type de données GEOJSON

Pour le type de données nous avons choisi GEOJSON, c'est le plus adapté pour ce cas d'utilisation. C'est un format qui permet de créer plusieurs types de structures tels que des lignes, des polygones, des chaînes de caractères ou encore des points, ce qui sera dans notre cas, le seul type de structure utilisé. Chaque objet GEOJSON possède donc un type de structure et des coordonnée GPS. A cet objet il est possible d'ajouter des propriétés de n'importe quel type, ce qui rend le format vraiment intéressant pour de futures améliorations.

IV. Conclusion

Pour conclure, nous allons vous rappeler les résultats principaux du projet AQROPOL. Il s'agit de connaître la pollution de la ville de Rennes.

Notre projet peut être appliqué dans la majorité des contextes de récolte de données qui utilisent des capteurs avec un budget limité. Il suffit simplement pour cela de changer le type de valeur des données récupérées par les capteurs. Par exemple nous pouvons utiliser notre projet pour mesurer la qualité de l'eau du robinet ou d'un lac en utilisant un capteur

adapté.

Nous envisageons des perspectives et des améliorations possibles pour ceux désirant reprendre celui-ci. Voici une liste non exhaustive des perspectives futures.

- Nous pourrions implémenter dans l'application Android:
    1. Un système de récompense en fonction de la quantité de données envoyées vers le serveur(le nombre de fois de mesures).
    2. Ouvrir la possibilité de se connecter à partir de l'application avec les réseaux sociaux pour partager la qualité de l'air mesurée.
    3. Affichage de statistiques comme le nombre de capteurs visités.
    4. Permettre l'affichage des mesures récupérées directement sur le NUC ainsi qu'un espace d'administration.

- Ajouter différents types de visualisation des données comme par exemple des barres inspirés des graphes en bâton avec pour hauteur la valeur de pollution mesurée, ou afficher des maps de styles différentes.
- Pouvoir afficher les valeurs mesurées d'un point précis quand l'utilisateur passe sa souris dessus.
- Proposer une implantation avec "Reactiv-Streams" pour gérer le nombre important de données à récupérer, et permettre d'observer les données des capteurs en temps réel.
- Proposer une interface de monitoring du NUC.

Comme le montre la liste d'amélioration au dessus, le projet AQROPOL peut être amélioré, mais rappelons également que c'est un projet modulable.