Cazalis Pauline – von Tolkacz Karol – Joret Eddy

Ville connectée

Pour une amélioration de la qualité de vie dans la ville de Pau

Rapport projet tutoré



Table des matières

[Introduction 2](#_Toc71127128)

[1 - Présentation du projet 3](#_Toc71127129)

[1.1 - Contexte 3](#_Toc71127130)

[1.2 – Problématique 3](#_Toc71127131)

[1.3 - Description du projet 4](#_Toc71127132)

[2 - Analyse du projet 5](#_Toc71127133)

[3 - Présentation du matériel 6](#_Toc71127134)

[3.1 – Arduino 6](#_Toc71127135)

[3.2 – Capteurs 7](#_Toc71127136)

[3.3 - Logiciels et environnement de travail 8](#_Toc71127137)

[4 - Déroulement du projet 9](#_Toc71127138)

[4.1 - Planning prévisionnel 9](#_Toc71127139)

[4.2 - Répartition des taches 10](#_Toc71127140)

[4.3 - Cahier des charges 10](#_Toc71127141)

[5 - Présentation du travail réalisé 11](#_Toc71127142)

[5.1 - Gestion des poubelles 11](#_Toc71127143)

[5.2 - Gestion de l’éclairage public 14](#_Toc71127144)

[6 - Difficultés rencontrées 15](#_Toc71127145)

[6.1 - Retard sur le planning initial 15](#_Toc71127146)

[6.2 - Difficultés techniques 15](#_Toc71127147)

[6.3 – Changements apportés au projet 15](#_Toc71127148)

[Conclusion 15](#_Toc71127149)

[Bilan des attentes 15](#_Toc71127150)

[Evolution de la solution 15](#_Toc71127151)

[Apports personnels 15](#_Toc71127152)

[Sources (référencé dans le rapport également pas juste a la fin) 16](#_Toc71127153)

[Annexes 16](#_Toc71127154)

# Introduction

Durant notre second semestre de Master Technologie de l’Internet à l’université de Pau et des Pays de l’Adour, nous avons été amenés à réaliser un projet tutoré dans le cadre de notre module « Gestion et réalisation de projet ».

Ce projet devait être réalisé en groupe. Les membres de notre groupe sont :

* Pauline CAZALIS
* Eddy JORET
* Karol VON TOLKACZ

Il a été proposé et encadré par Fabien LAMAS, Nicolas DUBOIS, Mikaël LEFEVRE, Guillaume FAUVET et avec l'aide de Thomas ANSELMI, Lamia AMRANI et Lucas PAUZIES, des ingénieurs logiciels de Capgemini (entreprise de services du numérique français basé à Pau).

Ce projet a pour but de partir d’une solution déjà existante d’agence connectée et d’étendre ce système à la ville de Pau pour améliorer sa qualité de vie ainsi que sa dépense énergétique.

Nous avons été deux groupes à travailler autour de ce projet. Le premier a concentré son travail sur l’urbanisme de la ville. Notre groupe, lui, s’est orienté sur l’environnement.

Nous avons donc commencé la mise en place de notre solution en respectant la problématique suivante : Quels changements pouvons-nous apporter pour améliorer la qualité de vie de Pau ?

Afin de structurer notre rapport écrit, nous allons dans un premier temps faire une présentation globale de notre projet, nous ferons ensuite la description du matériel utilisé et enfin nous présenterons de façon détaillé le travail réalisé et les difficultés rencontrées.

# 1 - Présentation du projet

# 1.1 - Contexte

Chaque année en France, un habitant produit en moyenne plus de 500kg de déchets. Malheureusement, ce nombre continu d’augmenter au fil des années.

A cause de la crise sanitaire, la vente de repas à emporter a considérablement augmenté.

De ce fait, les déchets s’entassent dans les poubelles de la ville, obligeant les passants à essayer de trouver une place pour jeter leurs déchets. Certains n’hésitent pas à les poser à côté des poubelles ce qui a pour effet de gâcher le paysage de la ville.

A l’heure actuelle aucune solution n’est mise en place pour remédier à ce problème.

Concernant l’éclairage public de la ville de Pau, en 2009 la consommation annuelle de l’éclairage public était de l’ordre de 2,9 mégawatts. Cela correspond à une production de CO (monoxyde de carbone) de 1518 tonnes.

En 2017, la ville a décidé de réduire sa consommation électrique en changeant les ampoules actuelles par des LED.

Il est donc primordial de pouvoir gérer ces deux cas en temps réel afin de pouvoir améliorer la qualité de vie des habitants tout en réduisant les coûts engendrés par la ville.

# 1.2 – Problématique

Ce projet, répartis entre les deux groupes, avait pour objectif la mise en place d’une « Ville Connectée » ou « Ville intelligente » c’est-à-dire une ville où l’utilisation de capteurs permet la récupération de données. Son objectif est l’amélioration de la qualité des services urbains et la réduction de leurs couts.

Ainsi notre groupe, avait deux tâches principales à réaliser :

* Gestion des déchets : détecter si une poubelle est remplie ou non
* Gestion de l’éclairage public : allumer ou éteindre des lampadaires en fonction de la luminosité extérieure

# 1.3 - Description du projet

Nous avons repris le projet proposé l’année dernière. Il consistait en la création d’une agence, équipé d’une solution de surveillance de données, se voulant être une prémisse à l’élaboration d’une solution à l’échelle d’une ville.

Celle-ci nous a permis de construire la base de notre projet tout en apportant des changements et des améliorations afin de pouvoir répondre à notre problématique.

Ainsi, pour pouvoir extraire et interpréter les données voulus, nous avions à disposition une carte Arduino ainsi que différents capteurs et logiciels, servant à récupérer, traiter et stocker ces données.

# 2 - Analyse du projet

# 2.1 - Cahier des charges

Contexte :

Ce projet est un POC (proof of concept) ayant pour objectif futur de pouvoir répondre à des opportunités de Ville Connectée avec la ville de Pau ou des alentours.

Objectifs :

Les objectifs de celui-ci sont :

* Récupérer les données des différents capteurs
* Mettre en place une architecture pour analyser les données récupérées
* Afficher les données analysées sur une page web

Contraintes :

* Contraintes techniques :

Tous les mardis, nous avions une réunion en ligne avec nos encadrants. Celle-ci nous permettait de discuter du projet afin de parler de nos avancées mais aussi de nos problèmes rencontrés. A l’issu de ces réunions un planning pour la semaine suivante était mis en place.

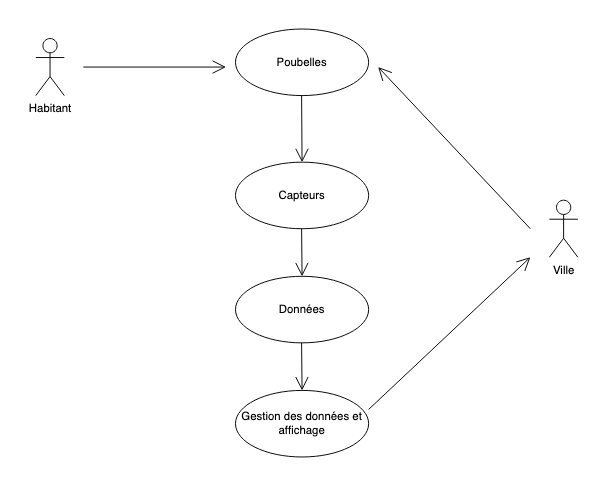
A cause de la crise sanitaire nous n’avons pas pu travailler ou nous voir en personne. Nous avons donc travaillé chacun de notre côté avec nos propres ordinateurs.

* Contraintes de temps :

Nous devions finir le projet et le rapport pour le 12 mai, ainsi que préparer une soutenance orale pour le 25 Mai.

## 2.2 – Analyse des données

Dans cette partie nous allons présenter nos deux tâches principales à l’aide de diagrammes de cas d’utilisation.

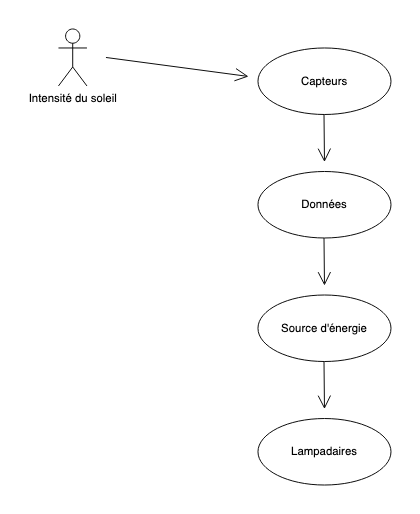


*Diagramme de cas d’utilisation de la gestion des poubelles*

Nous avons deux acteurs principaux :

* Les habitants de la ville qui se contentent de placer leurs déchets dans les poubelles
* La ville, qui gère les poubelles de la ville et regarde les données affichées.

Chaque poubelle possède un capteur de pression attitré, qui récupère la pression des déchets contenues à l’intérieur. Celui-ci envoie les données qui seront ainsi affichés et cela permettra à la ville d’agir en conséquence.



*Diagramme de cas d’utilisation de la gestion de l’éclairage public*

Dans ce diagramme nos capteurs récupèrent l’intensité lumineuse du soleil suivant le passage jour/nuit ou inversement. Cet évènement fournira une certaine donnée qui agira sur les lampadaires en les éteignant ou en les allumant.

# 3 - Présentation du matériel

Dans le but de répondre au mieux aux besoins techniques de notre projet, l’utilisation de plusieurs outils a été nécessaire. Ceux-ci sont présentés dans cette partie.

# 3.1 – Arduino

Nous disposons d’une carte Arduino NodeMCU ESP8266 de la marque Velleman, avec son câble d’alimentation USB/Micro-USB.

C’est une carte avec circuit intégré à un microcontrôleur avec connexion Wi-Fi, composé de 10 broches GPIO (General Purpose Input/Output), que l’on utilisera avec le langage C++ avec l’IDE Arduino.

Chaque GPIO peut être PWM (Pulse Width Modulation), I2C (Inter-Integrated Circuit) ou encore 1-Wire (adressage et utilisation de plusieurs capteurs à communication série sur une seule et même broche numérique).

Elle est aussi composée de deux boutons poussoirs rst (reset, redémarrage de la carte) et flash (permet de lancer une séquence de flasharge de la mémoire) et d’une antenne PCP qui permet d’améliorer la portée.



Arduino NodeMCU ESP8266 Velleman

Cette carte nous permet de recevoir des informations fournis par un ou plusieurs capteurs et de les transmettre vers un logiciel externe.

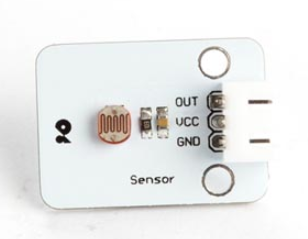
L’Arduino ESP8266 étant équipée d’un module wifi, cela nous permettra d’envoyer les données reçus (sous un format type -> Précision du format) via une connexion wifi, bien plus pratique qu’une connexion filière à l’échelle d’une ville.

Nous disposons aussi de 21 câbles pour la connexion de l’Arduino aux capteurs.

# 3.2 – Capteurs

3 capteurs nous ont été fournis pour obtenir les données voulus :

* 1x capteur photosensible qui utilise une photorésistance pour produire une tension de sortie.

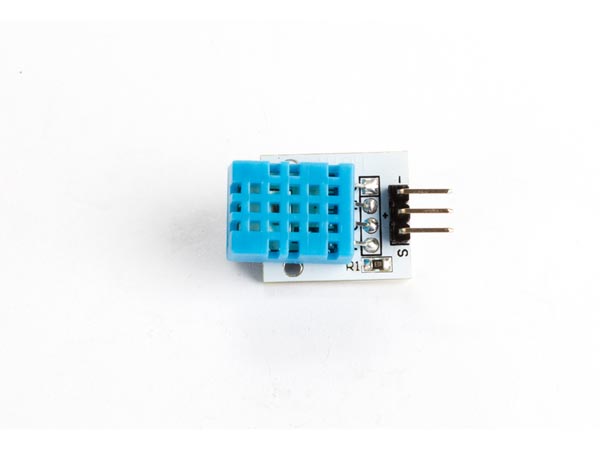


Capteur photosensible Velleman

Une photorésistance est une résistance dont la valeur change en fonction de la lumière qu’elle reçoit.

Grâce à ce capteur, nous allons pouvoir récupérer la luminosité actuelle de la ville et et donc gérer l’éclairage public.

* 2x capteurs de température (NTC) et d’humidité (DHT11).



Capteur de température et d'humidité digital DHT11 Velleman

Il va nous permettre de récupérer la température et l’humidité présente dans l’air de la ville.

# 3.3 - Logiciels et environnement de travail

Afin de pouvoir exploiter et afficher nos résultats, nous avons utilisé plusieurs logiciels :

Arduino IDE : c’est un logiciel de programmation pour la carte Arduino. Il va nous permettre de créer le code pour faire fonctionner la carte avec les différents capteurs.

* Broker MQTT Mosquitto : c’est un protocole de messagerie de type publish-subscribe. Il va permettre à notre Arduino d’envoyer et/ou de recevoir des messages en se connectant à un « topic » pour éviter que les messages ne se mélanges pas.
* Node-RED : c’est un outil de programmation, basé sur un navigateur, qui permet de développer des objets connectés à l’aide de flux. Il va nous permettre de simuler les données reçues pour la gestion des poubelles par un capteur fictif.
* MongoDB Atlas : c’est une base de données distribuée. Elle nous permet de stocker les données reçues par les différents capteurs pour pouvoir ensuite avoir un historique de celles-ci.
* Express JS : C’est un framework (permet de simplifier le travail des développeurs en offrant une architecture « prête à l’emploi ») qui fournit les fonctionnalités d’applications web et mobile fondamentales. Il est basé sur Node Js (environnement d’exécution qui permet d’utiliser JavaScript côté serveur).

Il va nous permettre de recevoir les données des capteurs et les traiter pour pouvoir les envoyer à la fois sur notre page web et sur la base de données pour l’historique.

* Angular : C’est un framework, écrit en JavaScript, qui permet la création d’applications web. C’est grâce à lui que nous allons pouvoir afficher nos données pour qu’elles soient visibles de l’utilisateur.
* Docker : C’est une plateforme logicielle qui permet de concevoir, tester et déployer des applications rapidement grâce à des conteneurs logiciels. Les conteneurs rassemblent tous les éléments nécessaires aux logiciels (bibliothèques, outils systèmes…) afin de pouvoir les exécuter dans n’importe quel environnement.

Ainsi, nous avons pu regrouper notre Broker MQTT, Node-RED ainsi que Express pour qu’ils puissent s’exécuter en même temps et correctement.

Pour pouvoir organiser notre projet, nous avons utilisé deux logiciels :

* Github : c’est un service web d’hébergement et de gestion de développement de logiciel. Il nous a permis de stocker nos différents fichiers en ligne afin que tous les membres du groupe puissent y avoir accès.
* Trello : c’est un outil de gestion de projet en ligne. Nos encadrants ont pu lister les différentes tâches que nous avions à faire pour la réalisation de ce projet.

Nous avons pu, grâce à cet outil, nous organiser pour la réalisation de ce projet.

# 

# 4 - Déroulement du projet

# 4.1 - Planning prévisionnel

Pour réussir notre projet tutoré, il nous fallait absolument passer par une planification efficace. Celle-ci a été mise en place via un planning prévisionnel ; ce document de travail permet de faire le lien entre les objectifs du projet et la façon dont il sera décliné de manière opérationnelle.

Le planning prévisionnel est un outil de travail qui nous permet de visualiser la manière dont le projet va s’organiser dans le temps et les différentes étapes à suivre pour atteindre les objectifs. Il permet de construire et planifier les étapes nécessaires à la réalisation du projet et de les visualiser dans le temps.

Pour le modéliser nous avons utilisé le logiciel *GanttProject*, qui nous permet de créer des diagrammes de Gantt.

Image planning prévisionnel

Image planning réel

# 4.2 - Répartition des taches

Après avoir établi ce planning, nous avons réparti ces différentes tâches entre nous trois.

Elles ont été partagées en fonction des compétences de chacun.

A détailler

* Ecriture du code pour les différents capteurs (Eddy Joret)
* Connexion MQTT de l’arduino à Node-RED (Eddy Joret)
* Simulation des données sur Node-RED (Eddy Joret)
* Création de docker (Karol von Tolkacz, Eddy Joret, Pauline Cazalis)
* Création d’un serveur Express (Node Js) (Pauline Cazalis)
* Récupération des données de MQTT au serveur ()
* Envoie des données du serveurs à base de données (Pauline CAZALIS, Eddy JORET)
* Création du dashboard sur Angular (Pauline Cazalis)
* Rédaction du rapport (Karol von Tolkacz, Joret Eddy, Cazalis Pauline)

# 5 - Présentation du travail réalisé

# 5.1 - Gestion des poubelles

Via la récupération d’informations concernant les poubelles de la ville, le premier travail consistait à afficher sur une carte les poubelles de la ville remplie ou non selon un code couleur précis.

Nous devions trouver une solution simple et efficace à cette demande. Nous avons donc décidé de placer un capteur de pression directement à l’intérieur de la poubelle.

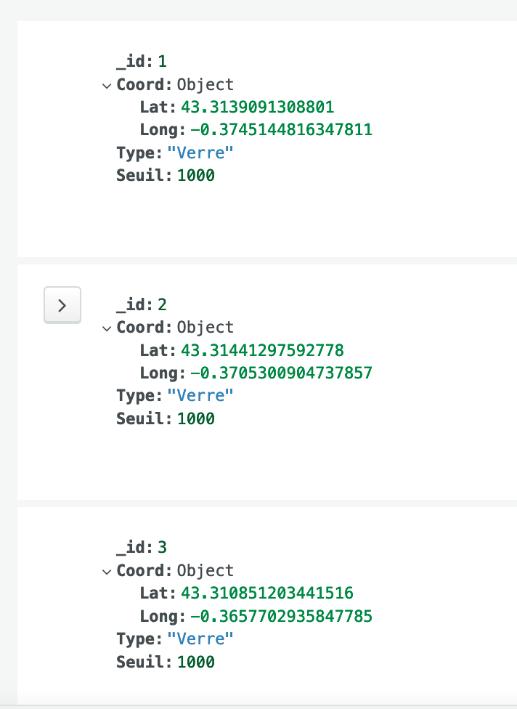
Ne possédant pas le capteur de pression pour l’Arduino, nous avons donc dû le simuler.

🡪 Création de la base de données

La ville possède 3 types de poubelles : verre, déchets communs et recyclables.

Les données qui nous sont intéresse sont les coordonnées de la poubelle, son type, son seuil, sa pression ainsi qu’un moyen d’identification. Pour cela nous avons créé deux types de collections :

* Poubelle\_id : contient les données globales des poubelles de Pau. Elles possèdent un id, des coordonnées, un type et un seuil.



Exemple du contenu de Poubelle\_id

* Poubelle\_hist : contient les données de temps réel des poubelles de Pau et est relié à la première collection grâce à la clef *Poubelle\_id*.



Exemple du contenu de Poubelle\_hist

🡪 Création du code de simulation

Détailler la simulation des données

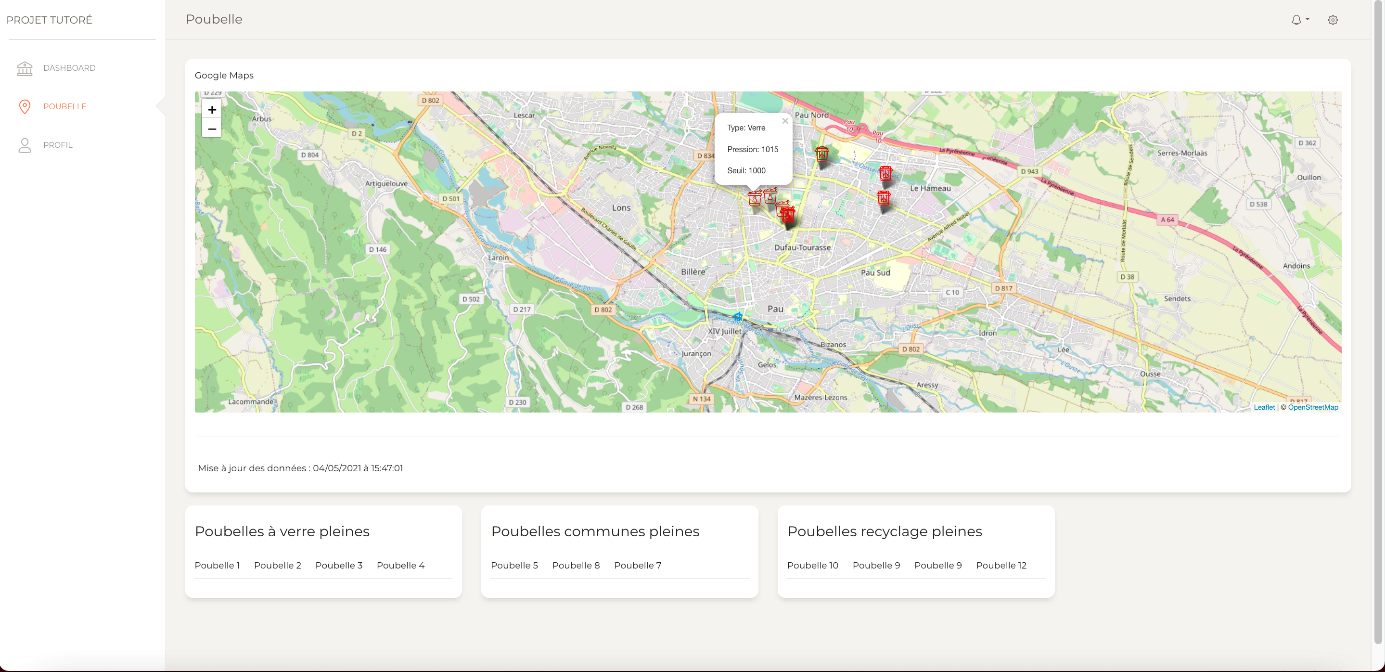
🡪 Création du serveur

Notre serveur nous permet de récupérer nos données via le broker MQTT puis va les sauvegarder dans la base de données et si besoin va fournir les données nécessaires à l’affichage.

Les données seront envoyées sur notre page web Angular via une socket. C’est un protocole réseau permettant la création d’un canal de communication à double sens entre un serveur et un navigateur.

🡪 Création de la page web

Après avoir reçu les données depuis la socket, Angular va stocker ces données pour pouvoir les afficher. Nous avons décidé de les afficher de façon séquentielle sur une carte (crée par la bibliothèque Leaflet). L’utilisateur pourra donc surveiller en temps réel l’évolution de celles-ci.



*Notre « dashboard » indiquant l’état en temps réel des poubelles de la ville de Pau.*

# 5.2 - Gestion de l’éclairage public

Notre deuxième tâche a été de proposer une solution pour réduire la consommation électrique et le cout concernant l’éclairage public de la ville de Pau.

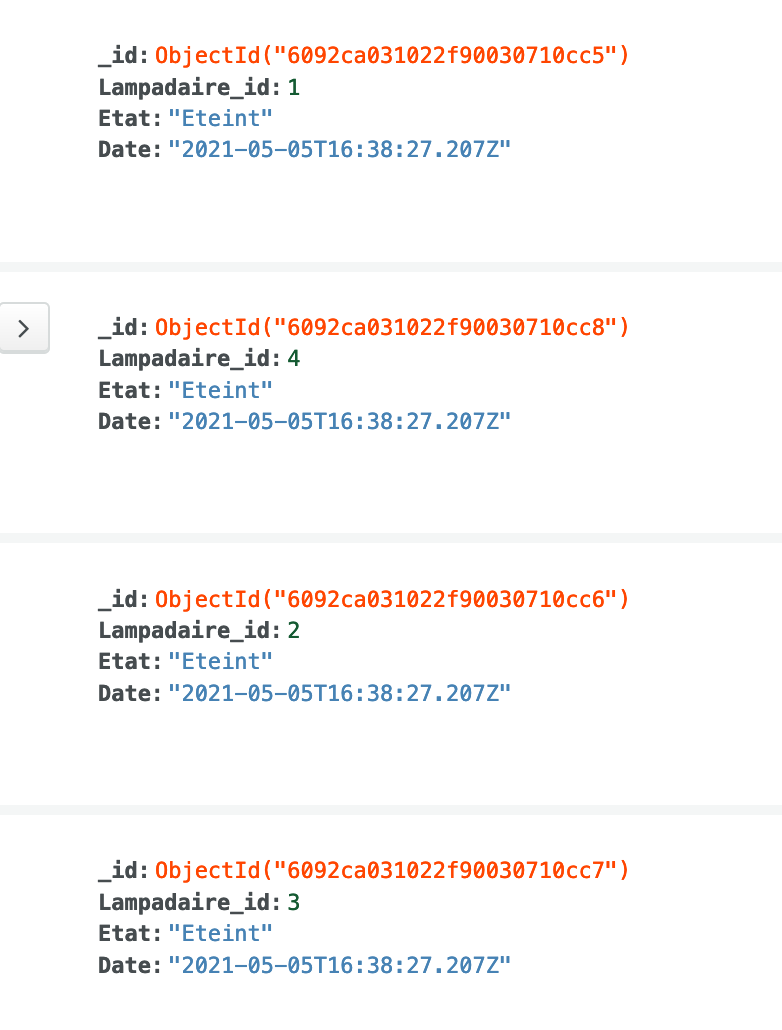
Nous avons opté pour la solution la plus simple : le capteur photo sensible va détecter s’il fait jour ou nuit de manière a activer ou non le lampadaire.

🡪 Création de la base de données

Notre base de données comporte deux collections :



Lampadaire\_hist



Screen à rajouter après remplissage

🡪 Création du code

(Eddy)

# 6 - Difficultés rencontrées

# 6.1 - Retard sur le planning initial

* Retard sur la gestion des poubelles
* Retard architecture BDD

# 6.2 - Difficultés techniques

* Problème de co entre mqtt et arduino
* Problème de co entre node-red et express (websocket)
* Repenser la base de données pour les poubelles

# 6.3 – Changements apportés au projet

* On a dû refaire plusieurs fois le code des poubelles, node-red et les sockets
* Changement de l’architecture

# Conclusion

## Bilan des attentes

Ce qu’on a pensé de ce projet

## Evolution de la solution

Qu’elles idées nous avons pour améliorer cette solution ?

Ca peut être intéressant de dire que votre code gérant l'éclairage public peut ouvrir des possibilités comme l'optimisation de l'éclairage avec la présence de capteurs de mouvements (passage de l'éclairage d'un mode normal à un mode veille et vice versa)

## Apports personnels

Ce que le projet nous a apporté

# Sources (référencé dans le rapport également pas juste a la fin)

* <https://www.larepubliquedespyrenees.fr/2010/11/13/un-meilleur-eclairage-pour-une-consommation-moindre,165788.php>)

# Annexes