- Équipe : 3
- Défi : 8
- Membres : Enzo, Idriss, Tristan, Audrey

Cahier des charges

Surveillance climatique d'une serre connectée

Nous sommes l'équipe 3 du projet H4ck The Farm. Notre défi consiste à mettre en place un système complet de surveillance climatique pour une serre, afin d'optimiser la croissance des cultures tout en réduisant la consommation énergétique et en facilitant le travail de l'agriculteur. L'ensemble du projet est réalisé collectivement.

Notre objectif est de collecter en temps réel les données climatiques internes à la serre, incluant la température, l'humidité, l'éclairement et le taux de CO_2. Nous souhaitons enrichir ces données par des prévisions météorologiques afin de prendre des décisions automatisées comme le déclenchement du chauffage ou l'ouverture de la ventilation. En parallèle, nous mettrons en place un système d'alerte basé sur Telegram, permettant une réception immédiate des notifications par l'agriculteur sur son smartphone ou ordinateur. Nous afficherons toutes ces données sur un tableau de bord web et conserverons un historique des mesures et des actions.

Nous prévoyons également de réaliser une cartographie de la réception LoRaWAN avec au moins vingt points de mesure afin d'évaluer la qualité du signal dans la zone de déploiement. Cela nous permettra de vérifier que les capteurs communiquent correctement avec la passerelle, d'identifier d'éventuelles zones à faible couverture, et de valider la fiabilité du réseau pour un usage agricole. Cette cartographie contribuera à garantir un fonctionnement stable du système dans différentes conditions et emplacements autour de la serre.

I - Architecture technique prévue: comment on va le faire techniquement.

Les capteurs LoRaWAN installés dans la serre mesurent la température, l'humidité, le taux de CO_2 et l'éclairement. Ils envoient ces données par ondes radio basse fréquence vers une antenne LoRaWAN installée à proximité. Cette antenne transmet ces données *via* Internet au cloud TTN (The Things Network), qui joue le rôle de réseau intermédiaire et de broker MQTT distant. Nous ne disposons pas de notre propre broker dans la VM, les données sont donc directement collectées depuis TTN *via* un script Python.

Dans la VM, nous hébergerons une base de données InfluxDB où seront stockées les mesures des capteurs, les actions réalisées, et les prévisions météo. Le serveur web développé avec Flask permettra d'afficher ces données, d'interagir avec les actionneurs, et d'accéder à un historique structuré.

Nous prévoyons d'intégrer à notre projet une interface web modélisée sur Draw.io. Elle servira de base à l'organisation du site de visualisation et permettra de justifier nos choix UX et UI. Cette interface représentera une serre avec une seule zone centrale instrumentée, où tous les capteurs sont placés à 1m50 de hauteur pour représenter au mieux la température moyenne. L'actionneur LoRaWAN permet de simuler une ouverture de bâche ou une ventilation lorsque certains seuils sont dépassés. En effet, comme nous ne pouvons pas actionner un vrai système physique dans ce cadre, nous utiliserons un dispositif de test simple, comme une diode LED ou une lumière de signalisation, pour simuler l'activation d'un actionneur (chauffage ou ventilation). Cette diode s'allumera ou s'éteindra lorsqu'un message MQTT « downlink » sera reçu *via* LoRaWAN, ce qui permettra de valider l'ensemble du circuit de commande, de la détection au déclenchement.

Un scénario prédictif sera envisagé pour simuler l'évolution de la maturation des plantes : par exemple, un taux de CO_2 élevé couplé à une température constante peut permettre d'anticiper une maturation dans trois jours, tandis qu'une aération trop précoce peut la retarder. Ces modélisations seront intégrées sous forme de règles métiers dans notre application web, afin de proposer une estimation visuelle du rendement attendu et de simuler les effets des actions à venir.

Nous intégrerons également une API météo permettant de récupérer les prévisions extérieures (température, humidité, précipitations, UV, vent, pression atmosphérique). Cette API sera interrogée régulièrement par un script Python dans la VM. Les données météo viendront compléter celles mesurées dans la serre afin d'affiner les décisions automatisées et de mieux anticiper les variations climatiques locales.

Une cartographie de la réception LoRa sera produite à partir d'au moins vingt relevés sur le terrain. Chaque point comportera des coordonnées GPS et un indicateur de qualité de signal. Cette carte nous permettra d'évaluer la couverture radio réelle autour de la serre et d'identifier d'éventuelles zones à faible connectivité.

Conformément aux consignes, nous utiliserons la passerelle LoRa eui-0080000000021b9e, située sur le toit de l'IUT, pour effectuer nos relevés dans la zone attribuée à notre équipe : rues face à la façade panneaux solaire (Sud).

L'ensemble de notre système constitue un jumeau numérique de la serre. Il représente en temps réel les conditions climatiques internes et les états des actionneurs, permet d'anticiper les évolutions à l'aide de scénarios métier et de prévisions météo, et offre des capacités de simulation et d'aide à la décision.

II - Fonctionnalités détaillées: ce que l'utilisateur pourra faire concrètement.

Nous partirons du principe que l'agriculteur accède à notre système *via* l'interface web hébergée dans la VM. En plus des données climatiques et des prévisions, l'interface présentera les messages d'alerte et les historiques des actions déclenchées automatiquement ou manuellement, ainsi que les relevés des capteurs. Cela permettra à l'utilisateur de suivre les évolutions dans le temps, de mieux comprendre les interventions passées, et d'analyser les corrélations entre les données et les réponses du système. Nous porterons une attention particulière au design de l'interface (UI/UX) pour qu'elle soit claire, intuitive et agréable à utiliser, même pour une personne sans compétence technique. Le site sera responsive, c'est-à-dire qu'il pourra s'adapter en fonction du support numérique utilisé pour consulter le site comme un smartphone.

Les commandes manuelles seront transmises *via* MQTT vers TTN, qui les relaiera au réseau LoRaWAN. Les actionneurs réagiront alors à ces ordres comme s'ils les avaient reçus du cloud directement.

Pour simuler un retour d'état réaliste, nous ajouterons un capteur de butée (ou son équivalent logiciel) afin de confirmer que les trappes virtuelles ont bien été ouvertes ou fermées. Cette confirmation sera affichée dans l'interface web, permettant de détecter toute anomalie (commande envoyée mais état inchangé).

Chaque actionneur pourra donc aussi être destinataire de messages *via* MQTT. Par exemple, l'ouverture d'une trappe (simulée) sera déclenchée si la température dépasse un certain seuil, ou si le taux de CO_2 est trop élevé. Un script surveillera aussi l'absence de réception des données pour déclencher une alerte à l'utilisateur.

Ces seuils seront définis dans un fichier de configuration Python que l'on pourra adapter facilement. Une évolution future envisagée est de permettre à l'utilisateur de modifier ces seuils directement depuis l'interface web, selon les cultures ou les saisons. Les données importantes seront mises en avant, les alertes bien visibles, et les actions manuelles accessibles simplement. Chaque commande manuelle envoyée *via* l'interface (chauffage ou ventilation) sera enregistrée dans la base de données avec un horodatage, le type d'action, et l'état de confirmation (*via* capteur de butée). Cela permettra de garder une trace complète des interactions utilisateur.

Notre API météo permettra notamment d'intégrer aux tableaux de bord des prévisions de température, d'humidité et d'ensoleillement pour les prochaines heures ou jours. Ces informations enrichiront le système de surveillance pour rendre les déclenchements d'actionneurs plus intelligents et moins réactifs uniquement aux conditions immédiates. Le script sera pensé pour éviter les appels excessifs et respectera les limites de la plateforme météo choisie.

Nous allons récupérer automatiquement les mesures envoyées par les capteurs depuis le broker MQTT de TTN. Ces messages seront ensuite relayés dans notre VM à travers un broker Mosquitto local. À partir de là, nous aurons un accès simplifié pour traiter, stocker et afficher les informations. Notre serveur web permettra de visualiser les données climatiques actuelles, l'historique des mesures, l'état des dispositifs contrôlés et d'interagir manuellement avec les actionneurs si nécessaire. Des scripts permettront de déclencher automatiquement certaines actions selon les données mesurées, comme l'allumage du chauffage ou l'ouverture de la ventilation. Les utilisateurs recevront des notifications Telegram en cas de dépassement de seuils définis. L'ensemble des mesures, actions et alertes sera consigné. L'interface web proposera également une fonctionnalité de téléchargement de l'historique complet des données climatiques et des actions au format CSV. Ce fichier pourra être ouvert dans un tableur (Excel ou équivalent) pour permettre des analyses personnalisées ou une sauvegarde hors-ligne. Nous ajouterons aussi une récupération des données météo.

III - Contraintes et choix techniques: ce que l'on a choisi en termes de technologie et pourquoi.

Tous les services nécessaires à notre projet seront hébergés dans une unique machine virtuelle. Nous ferons appel au broker MQTT de TTN pour recevoir les mesures des capteurs, et utiliserons un relais interne pour transférer ces messages vers un Mosquitto local, afin de mieux contrôler les échanges internes. Ce relais Mosquitto local nous permet de centraliser les flux MQTT dans la VM, facilitant le développement, le débogage et l'intégration des scripts sans dépendre constamment du cloud de TTN. Il constitue un point unique de traitement pour tous les messages entrants ou sortants.
Les technologies utilisées sont les suivantes : Python, Flask, InfluxDB, Mosquitto, Grafana. Nous mettons un point d'honneur à assurer l'interopérabilité entre les composants, et à rendre l'interface utilisateur la plus claire et fonctionnelle possible. Chaque étape technique est discutée et décidée en groupe.

Nous utiliserons un système de gestion de base de données (SGBD) adapté aux séries temporelles : InfluxDB. Ce SGBD nous permettra de stocker efficacement les données horodatées provenant des capteurs et des prévisions météo. Nous pourrons ainsi visualiser l'évolution des conditions climatiques dans le temps, générer des historiques, des alertes, et optimiser nos décisions automatiques.

Pour restreindre l'accès à certaines fonctionnalités sensibles de l'interface web, notamment les commandes manuelles et l'historique des actions, une authentification basique par identifiant et mot de passe sera mise en place. Cette couche de sécurité a pour objectif de protéger le système contre les accès non autorisés, même si le projet est hébergé localement.

IV - Planning prévisionnel: ce qu'on prévoit de faire chaque jour

Mardi 3 juin

- Relais MQTT TTN vers Mosquitto local fonctionnel
- InfluxDB installée et accessible
- Structure de base de l'application Flask créée (routes, vues, templates)
- Bot Telegram configuré et opérationnel (clé API, canal test)

Mercredi 4 juin

- Données capteurs récupérées *via* MQTT et insérées dans InfluxDB
- API météo intégrée avec parsing et insertion dans InfluxDB
- Première visualisation des données climatiques en temps réel *via* Flask
- Script de base pour alertes Telegram sur seuils simples

Jeudi 5 juin

- Affichage des prévisions météo dans l'interface
- État des actionneurs simulés visible dans le site
- Intégration des premières règles métier simples (ex. température > 30°C → ventilation)

Vendredi 6 juin

- Déclenchement automatique d'un actionneur simulé en fonction des données mesurées
- Vue historique des mesures activée avec possibilité d'export CSV
- Intégration du design Draw.io à l'interface (serre, capteurs, actionneurs)

Lundi 9 juin

- Tests complets des déclenchements automatiques (chauffage, ventilation)
- Interface finalisée avec affichage responsive pour ordinateurs et mobiles

Mardi 10 juin

- Relevés terrain LoRa réalisés sur au moins 20 points
- Génération de la carte de réception LoRa et intégration dans Flask

Mercredi 11 juin

- Tests finaux du système : MQTT, météo, automatisation, alertes
- Confirmation d'état de l'actionneur *via* capteur de butée ou équivalent
- Script de détection d'anomalies (absence de données) opérationnel

Jeudi 12 juin

- Vidéo de présentation du projet finalisée (démonstration complète)
- Vidéos individuelles des membres enregistrées
- Export de toutes les données et livrables techniques préparé

Vendredi 13 juin

- Rapport technique final remis
- Tableau récapitulatif des activités du groupe terminé
- Remise complète de tous les livrables (scripts, app Flask, base, vidéos, documents)

V - Livrables

Nous fournirons le cahier des charges, les scripts nécessaires (relais MQTT, récupération météo, alertes Telegram), l'application Flask, le tableau de bord Grafana, la base de données contenant les mesures capteurs, la cartographie LoRaWan, une vidéo de présentation du groupe et une vidéo individuelle par membre. Un rapport technique complet ainsi qu'un tableau récapitulatif des activités du groupe seront remis pour permettre la reproduction du projet par un tiers.

Vous trouverez ci-dessous la maquette présentant l'agencement du site web, ainsi que les différentes pages qui le composent :

Figure 1 : maquette de la page d'accueil

Cette maquette (figure 1) représente la page d'accueil du site proposé au client. Le premier élément est une carte de la serre contenant les capteurs, les informations sur la température, l'humidité, le taux de CO_2, etc seront affichées juste en dessous. Elles s'afficheront en temps réel puisque les données qui s'afficheront proviendront directement des informations envoyées par les capteurs de la serre. En dessous de ces mesures se trouvent alors des cadres de couleurs avec un label qui indiquera si le niveau de mesure est bon, moyen ou mauvais (respectivement vert, jaune/orange, rouge). Ensuite, en fonction de ces mesures, il y aura un champ prévision qui permettra de faire des prévisions sur la pousse des plantes, si la météo peut varier, comment s'adapter et autres informations qui aideront l'agriculteur. Finalement, il pourra choisir, si les mesures ne sont pas correctes, de les modifier : il pourra notamment augmenter ou diminuer la température de la serre, ouvrir ou fermer les ouvrants pour la ventilation et d'autres fonctionnalités.

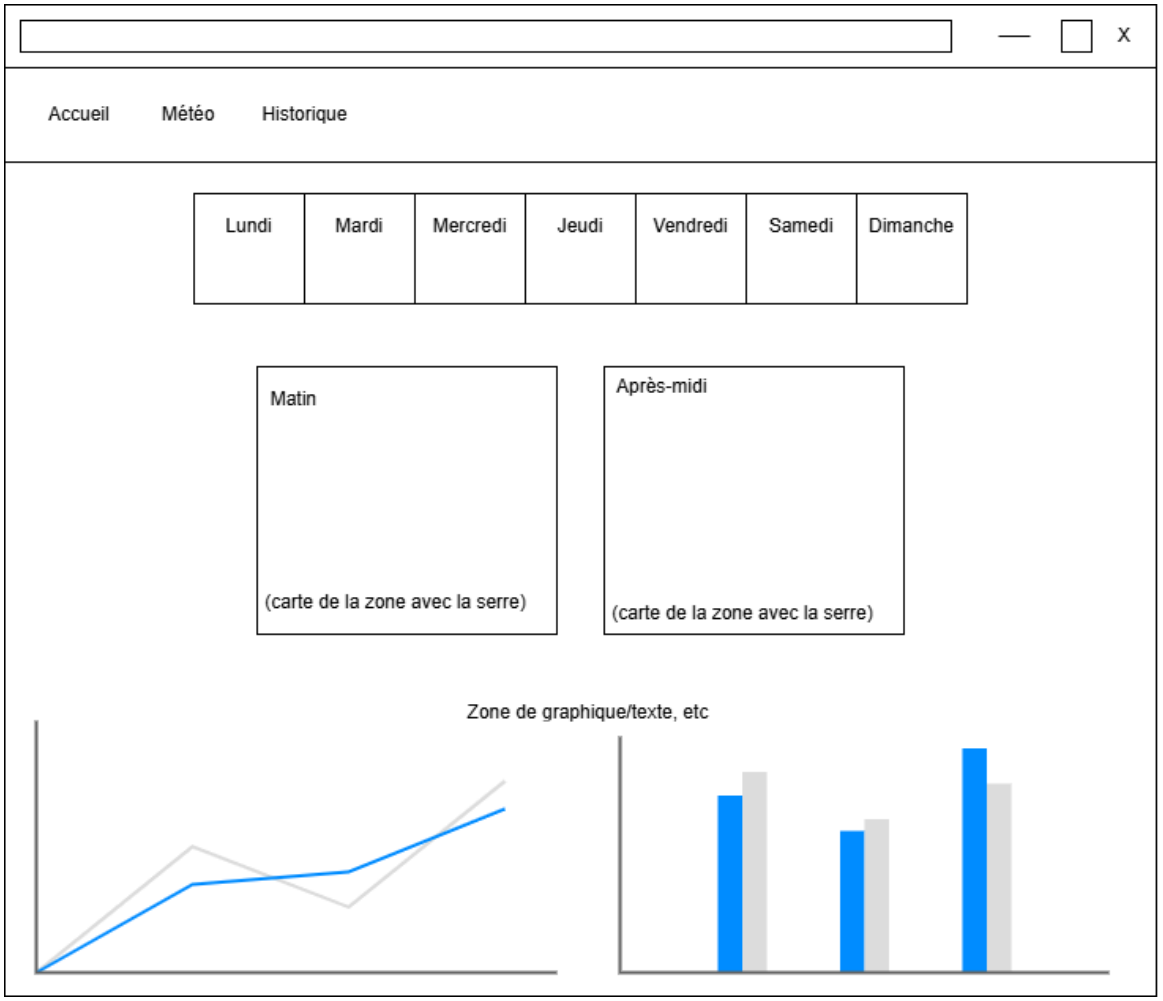

Figure 2 : maquette de la page météo

Cette maquette (figure 2) représente l'interface de l'onglet météo du site. En haut, un calendrier permet de sélectionner un jour de la semaine. Pour chaque jour, on peut consulter les conditions climatiques prévues le matin et l'après-midi, affichées sous forme de cartes météo détaillées. Sur ces cartes, un point de repère indique précisément l'emplacement de la serre, permettant d'avoir une vision claire de son environnement immédiat.

Juste en dessous, deux graphiques affichent l'évolution de plusieurs paramètres extérieurs sur plusieurs jours : la température, l'humidité et la vitesse du vent. Ces données permettent d'anticiper les variations météorologiques susceptibles d'influencer le climat intérieur de la serre et d'ajuster les réglages en conséquence.

figure 3 : maquette de la page historique

Cette page (figure 3) permet de consulter l'évolution des données relevées par les capteurs, telles que la température, l'humidité ou le taux de CO_2, sur différentes périodes. Le graphique peut être ajusté selon la période choisie, et un filtre permet de sélectionner le type de donnée à afficher. En dessous, un tableau présente l'historique des actions réalisées dans la serre, avec pour chaque ligne une description, la date et le capteur concerné. Toutes les informations sont extraites de la base de données interne.

Signatures: