



Université Mohamed Khider de Biskra
Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie
Département d'Informatique

Ordre N° :RTIC_Start Up_ /M2/2025

MÉMOIRE DE MASTER

Informatique
Spécialité : RTIC

Présenté et soutenu par :

DJOUDI Oussama Abdelilah – REZEG Mohamed Selimane

Système Intelligent Basé sur l'IoT et Techniques de l'IA pour la Supervision de la Culture Aquaponique (SISCA)

Jury :

DR.	SAOULI Rachida	Prof.	Université de Biskra	Président
DR.	TERRISSA Sadek Labib	Prof.	Université de Biskra	Rapporteur
DR.	BOURAKECHE Samir	MCB	Université de Biskra	Examinatrice

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Remerciements

Tout d'abord, je tiens à remercier **Dieu**, le Tout-Puissant, qui m'a accordé la réussite et la clairvoyance à chaque étape, et m'a inspiré la patience et la force pour accomplir cet accomplissement.

Je tiens à exprimer ma profonde gratitude à Monsieur le Professeur **Terrisa Sadek Labib**, enseignant au Département d'Informatique de la Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie à l'Université Mohamed Khider Biskra, pour avoir accepté de diriger ce travail. Son accompagnement constant, sa disponibilité, ses conseils avisés et ses remarques constructives ont été essentiels à la réalisation de ce mémoire. Sa rigueur scientifique et son engagement pédagogique ont constitué pour moi un véritable guide tout au long de cette recherche.

Je remercie également l'ensemble du corps enseignant et administratif du Département d'Informatique de l'Université Mohamed Khider Biskra pour la qualité de l'enseignement reçu tout au long de mon cursus, ainsi que pour leur soutien moral et académique.

Je tiens à exprimer ma reconnaissance envers mes collègues et camarades de promotion, dont les échanges enrichissants et le soutien mutuel ont contribué à rendre cette expérience encore plus formatrice.

Enfin, je remercie profondément ma famille, en particulier mes parents, pour leur amour, leur patience et leur soutien inconditionnel durant toutes mes années d'études.

À toutes et à tous, je vous adresse mes remerciements les plus sincères.

Dédicace

À mes parents, pour leur amour inconditionnel, leurs sacrifices et leur confiance qui ont toujours été le terreau de ma réussite.

À mon encadreur, Monsieur le Professeur **Terrissa Sadek Labib**, pour son soutien bienveillant, ses conseils éclairés et sa disponibilité tout au long de ce projet.

À mes camarades et coéquipiers de promotion, pour leur esprit de collaboration et les échanges riches qui ont jalonné notre parcours en Master 2.

À mes amis et à ma famille élargie, pour leur présence, leurs encouragements et les précieux moments de détente partagés, sans lesquels ce travail n'aurait pas la même saveur.

À tous ceux, proches ou lointains, qui, de près ou de loin, ont contribué à faire de cette aventure académique une expérience inoubliable.

Résumé

Ce mémoire propose la création d'un système intelligent appelé SISCA (Système Intelligent de Supervision et de Contrôle pour l'Aquaponie) pour répondre aux défis croissants liés à la sécurité alimentaire, à la raréfaction de l'eau et à la durabilité de l'agriculture. Il combine l'intelligence artificielle, le cloud computing et l'Internet des objets pour surveiller et optimiser les fermes aquaponiques en temps réel. SISCA collecte les données des capteurs (pH, température, OD, ammoniac, nitrates, etc.), les analyse à l'aide de modèles d'IA (GRU, RF), recherche les anomalies et prédit les conséquences environnementales. Grâce à son architecture SaaS multi-utilisateurs, SISCA offre à chaque exploitant agricole un accès individualisé, sécurisé et évolutif. Ce projet représente une véritable avancée vers une agriculture intelligente, résiliente et automatisée, adaptée aux contextes régionaux comme celui de l'Algérie.

Mots-clés : Aquaponie, Internet des Objets, Intelligence Artificielle, SaaS, GRU, RF, Agriculture intelligente, Surveillance de l'eau.

Absract

This memory suggests the creation of an intelligent system called SISCA (Système Intelligent de Supervision et de Contrôle pour l'Aquaponie) in response to the growing challenges associated with food security, water rarification, and agricultural durability. It combines artificial intelligence, cloud computing, and the Internet of Things to monitor and optimize aquaponic farms in real time. SISCA gathers sensor data (pH, temperature, DO, ammoniac, nitrates, etc.), analyzes it using IA models (GRU, RF), looks for anomalies, and predicts environmental consequences. Because of its multi-user SaaS architecture, SISCA provides each agricultural exploitant with individualized, secure, and evolving access. This project represents a real step toward intelligent, resilient, and automated agriculture that is tailored to regional contexts like Algeria.

Keywords : Aquaponics, Internet of Things (IoT), Artificial Intelligence (IA), SaaS, GRU, RF, Smart Agriculture, Water Quality.

ملخص

تقترح هذه الذاكرة إنشاء نظام ذكي يُسمى SISCA (النظام الذكي للإشراف والتحكم في الزراعة المائية) استجابةً للتحديات المتزايدة المرتبطة بالأمن الغذائي، وندرة المياه، واستدامة الزراعة. يجمع هذا النظام بين الذكاء الاصطناعي والحوسبة السحابية وإنترنت الأشياء لمراقبة وتحسين مزارع الزراعة المائية آنياً. يجمع SISCA بيانات الاستشعار (الرقم الهيدروجيني، درجة الحرارة، الأكسجين المذاب، الأمونيا، النتريت، إلخ)، ويحللها باستخدام نماذج الذكاء الاصطناعي (GRU و RF)، ويبحث عن أي شذوذ، ويتنبأ بالعواقب البيئية. بفضل بنيته البرمجية كخدمة (SaaS) متعددة المستخدمين، يوفر SISCA لكل مستغل زراعي وصولاً فردياً وآمناً ومتطوراً. يمثل هذا المشروع خطوة حقيقية نحو زراعة ذكية ومرنة وآلية، مصممة خصيصاً للسياقات الإقليمية مثل الجزائر.

الكلمات المفتاحية: الأكوابونيك، إنترنت الأشياء، الذكاء الاصطناعي، SaaS، GRU، RF، الزراعة الذكية، مراقبة جودة المياه.

Table des matières

Table des Matières

Table des Figures

Liste des Tableaux

Liste des abréviations

Introduction Générale	1
1 État de l'art	3
1.1 Introduction	3
1.2 L'aquaponie : principes et fonctionnement	3
1.2.1 Définition de l'aquaponie	3
1.2.2 Fonctionnement du système aquaponique	4
1.2.3 Avantages de l'aquaponie	4
1.2.4 Défis et considérations	5
1.2.5 Qualité de l'eau en aquaponie	5
1.2.5.1 Paramètres essentiels	6
1.2.5.2 Effets des dépassements des paramètres de qualité de l'eau	7
1.2.5.3 Les causes d'une mauvaise qualité d'eau en aquaponie	7
1.3 L'Internet des Objets (IoT)	8
1.3.1 Définition de l'IoT dans l'agriculture	8
1.3.2 Caractéristiques de l'IoT	9
1.3.3 Avantages de l'IoT en agriculture	10
1.3.4 Applications de l'IoT en agriculture de précision	10
1.3.5 Défis futurs de l'Internet des Objets (IoT)	10
1.3.6 Protocoles de l'IoT	11
1.3.7 Protocoles les plus courants	12
1.3.7.1 Protocole de Communication – MQTT	12
1.4 Intelligence Artificielle (IA)	13
1.4.1 Définition de l'intelligence artificielle (IA)	13
1.4.2 Avantages de l'IA en agriculture	14

1.4.3	Limites de l'IA en agriculture	14
1.4.4	L'avenir de l'IA en agriculture	15
1.5	Cloud computing	15
1.5.1	Définition Cloud computing	15
1.5.2	Caractéristiques du Cloud Computing	16
1.5.3	Les types de Cloud Computing	17
1.5.4	Modèles de services du Cloud Computing	18
1.5.4.1	SaaS – Software as a Service	19
1.5.4.2	PaaS – Platform as a Service (Plateforme en tant que Service)	20
1.5.4.3	IaaS – Infrastructure as a Service (Infrastructure en tant que Service)	21
1.5.5	Défis de l'intégration du Cloud Computing et de l'IoT	21
1.6	Travaux similaires sur l'aquaponie intelligente	23
1.7	Conclusion	24
2	Conception du Système SISCA	25
2.1	Introduction	25
2.2	Architecture Générale du Système	25
2.2.1	Description Globale du Système	25
2.2.2	Composants matériels principaux	27
2.2.2.1	Capteurs (sensors)	27
2.2.2.2	Microcontrôleur(s)	30
2.2.2.3	Actionneurs	31
2.2.2.4	LED horticole	32
2.2.2.5	Caméra de surveillance	32
2.2.3	Fréquence d'Échantillonnage	33
2.3	Analyse des Besoins	33
2.3.1	Besoins Fonctionnels	33
2.3.2	Besoins Non Fonctionnels	34
2.4	Modélisation UML du Système	35
2.4.1	Diagramme de cas d'utilisation	35
2.4.2	Diagramme de classes	37
2.4.3	Diagramme de Séquence	38
2.5	Conclusion	41
3	Implementation	42
3.1	Introduction	42
3.2	Langages de programmation	42
3.3	Outils de développement	44
3.4	Implémentation du modèle prédictif avec le Deep Learning	44
3.4.1	Définition du Deep Learning	45
3.4.2	Modèle Utilisé	45

3.4.2.1	Modèle principal : GRU (Gated Recurrent Unit)	45
3.4.2.2	Modèle secondaire : Random Forest Regressor (RF)	45
3.4.3	Méthodologie	46
3.4.4	Résultats expérimentaux : comparaisons des valeurs réelles et prédites	47
3.4.4.1	Comparaison Réelles vs Prédites — pH	47
3.4.4.2	Comparaison Réelles vs Prédites — Température — Turbidité	48
3.4.5	Analyse des performances du modèle de prédiction	48
3.4.6	hyperparamètres du modèle IA	49
3.4.7	Fonctionnement du Système SISCA	51
3.4.7.1	Quelques fonctions du système	52
3.5	Structure de la base de données	54
3.5.1	Table des utilisateurs (users)	54
3.5.2	Table des règles capteurs (sensor_rules)	54
3.5.3	Table des relevés de capteurs (sensor_readings)	54
3.5.4	Table des pompes/actionneurs (pumps)	54
3.5.5	Table des notifications (notifications)	54
3.6	Présentation de la plateforme	55
3.6.1	Interface utilisateur et modules fonctionnels	55
3.7	Conclusion	69
Conclusion Générale		70

Table des figures

1.1	Cycle de l'aquaponique [4]	4
1.2	Image description pour l'IoT [11].	9
1.3	Protocoles IoT [17].	12
1.4	Exemples de topologies MQTT [18].	13
1.5	Présentation du Cloud Computing [24].	16
1.6	Modèle de Cloud Computing [27].	19
1.7	Architecture multi-tenant [29].	20
1.8	Scénarios d'application pilotés par le paradigme CloudIoT et défis associés [31].	22
2.1	L'architecture générale du système SISCA.	26
2.2	L'architecture d'unité extérieure système SISCA.	26
2.3	L'architecture d'unité intérieure système SISCA.	27
2.4	Le capteur DS18B20.	28
2.5	Sonde pH analogique.	28
2.6	Capteur analogique EC/TDS.	28
2.7	Sonde DO analogique.	29
2.8	Capteur turbidity analogique.	29
2.9	Ammoniac NH_3 .	29
2.10	Nitrite.	30
2.11	Arduino Uno.	30
2.12	ESP8266.	30
2.13	Breadboard.	31
2.14	Pompes à eau et à air.	31
2.15	Vannes solénoïdes.	32
2.16	Servo moteur.	32
2.17	Diagramme de cas d'utilisation du système SISCA.	35
2.18	Diagramme de classes du système SISCA.	37
2.19	Diagramme de séquence traitement des demandes d'accès.	39
2.20	Diagramme de séquence surveillance et interaction utilisateur.	40
3.1	Architecture GRU.	45
3.2	Préparation et nettoyage des données.	46

3.3	Méthodologie de modèle.	47
3.4	Comparaison Réelles-Predites.	47
3.5	Comparaison Réelles-Predites.	48
3.6	Trois indicateurs performances du modèle.	49
3.7	Fusion GRU + Random Forest.	50
3.8	Fonctionnement d'IA in SISCA.	51
3.9	Fonction Configuration des composants, des pins et des intervalles Points.	52
3.10	Fonction Envoi des données du capteur d'eau.	53
3.11	Fonction Lecture des commandes moteur depuis le serveur.	53
3.12	Home page.	55
3.13	Register page.	56
3.14	Page admin.	56
3.15	Status et action des comptes.	57
3.16	Page dashboard-fish.	57
3.17	Page dashboard-plant.	58
3.18	Visualisation des paramètres.	59
3.19	Actionneur de SISCA.	60
3.20	Page information et statistiques.	61
3.21	Génération de statistiques.	62
3.22	Page Analytiques.	63
3.23	Page Détection des Anomalies dans les Prédictions.	64
3.24	Page rules.	65
3.25	Interface de gestion des pompes.	66
3.26	Page settings.	66
3.27	Page en arabe.	67
3.28	Gestion du mot de passe.	68
3.29	Page de notifications.	68

Liste des tableaux

1.1	Paramètres qualité de l'eau pour l'aquaculture et pour l'hydroponie.	6
1.2	Paramètres essentiels de l'eau en aquaponie.	6
1.3	Effets des dépassements des paramètres de qualité de l'eau.	7
2.1	Fréquence d'Échantillonnage.	33

Liste des abréviations

SISCA :	Système Intelligent de Supervision et de Contrôle en Aquaponie
IoT :	Internet des Objets
IA / AI :	Intelligence Artificielle / Artificial Intelligence
RF :	Random Forest Regressor
GRU :	Gated Recurrent Unit (variante de LSTM)
MQTT :	Message Queuing Telemetry Transport
ESP8266 :	Microcontrôleur Wi-Fi utilisé pour la transmission IoT
DB :	Base de données
UI :	Interface utilisateur (User Interface)
SaaS :	Software as a Service (logiciel en tant que service)

Introduction Générale

En raison de la croissance démographique, de l'épuisement des ressources naturelles et du changement climatique, l'agriculture durable est devenue une priorité mondiale. Parmi les solutions innovantes, l'aquaponie se distingue par une technique respectueuse de l'environnement qui associe l'élevage de poissons (aquaculture) à la culture de plantes en terre (hydroponie). Ce système repose sur un équilibre biologique où les déchets de poissons, riches en nutriments, servent d'engrais aux plantes, qui purifient à leur tour l'eau pour les poissons. Malgré ses nombreux avantages économiques et écologiques, ce modèle reste vulnérable. En effet, l'aquaponie traditionnelle souffre d'un manque d'automatisation, de supervision intelligente et de capacités d'anticipation face aux déséquilibres environnementaux. Ces défauts peuvent entraîner une mauvaise qualité de l'eau, des pertes dues à l'érosion ou une faible productivité végétale.

La rapide croissance des technologies numériques rend possible aujourd'hui de penser à une gestion efficace d'une manière automatisée et prévisible, grâce à la combinaison d'objets connectés, d'intelligence artificielle et de services en nuage, qui repose sur le suivi en temps réel de données importantes, détection d'erreurs invisibles à l'œil nu, conseils pour actions préventives automatiques. Selon le contexte technologique et environnemental nous construisons notre projet SISCAs (Système Intelligent de Supervision et Contrôle dans Aquaponie) autour d'une plateforme intelligente accessible par web intégrant des capteurs connectés ; algorithmes avancés et interface utilisateur.

L'objectif principal du projet SISCAs est donc de créer une solution intelligente, basée sur le cloud, pour la supervision avancée des systèmes aquaponiques. Plus précisément, le système doit être capable de garantir une collecte automatisée de données en temps réel, de fournir un tableau web interactif pour la visualisation et le contrôle, d'intégrer des modèles d'intelligence artificielle pour la détection des anomalies et la prévision de l'évolution future, et de déployer une architecture multi-utilisateurs basée sur le modèle SaaS. Ce projet privilégie une architecture modulaire, distinguant clairement l'unité cloud interne (base de données, serveur web, interface graphique et IA) de l'unité physique externe (capteurs + microcontrôleur).

L'étude de la façon de faire des choses montre que les solutions disponibles en aquaponie connectée sont souvent limitées à la simple collecte d'informations sans vraie analyse intelligente ni gestion personnelle. Beaucoup de projets ouverts ou commerciaux ne donnent pas des fonctionnalités avancées pour organiser, prédire ou analyser de plusieurs facteurs. Le projet SISCAs remplit ce manque en mettant, dès son départ, une intelligence capable de saisir des relations complexes entre paramètres d'eau, trouver déséquilibres possibles, et proposer action préventive. Aussi, sa base SaaS aide à l'adapter facilement à plusieurs utilisateurs et à différents milieux ruraux.

La méthodologie adoptée repose sur une approche technico-fonctionnelle articulée autour de plusieurs axes : l'intégration de capteurs via microcontrôleur, la transmission des données par protocole MQTT, le stockage dans une base MySQL hébergée sur serveur cloud, l'affichage dans un tableau de bord web développé en HTML/CSS/JS, et enfin, l'implémentation d'algorithmes d'IA (GRU) pour l'analyse et la prédiction. Le système est conçu de manière évolutive, modulaire, et orientée utilisateur.

Ce mémoire est structuré en plusieurs chapitres. Le premier chapitre présente les concepts de base liés à l'aquaponie, à l'IoT, à l'IA et aux architectures SaaS. Le deuxième chapitre est consacré à la conception technique du système SISCA, en détaillant l'architecture, les choix matériels et logiciels, ainsi que la modélisation UML. Le troisième chapitre aborde l'implémentation, les tests et la validation de chaque composant du système. Enfin, une conclusion générale viendra synthétiser les apports du projet et proposer des pistes d'amélioration pour les versions futures de SISCA.

Chapitre 1

État de l’art

1.1 Introduction

L’agriculture intelligente progresse rapidement grâce à l’intégration des technologies innovantes tels que l’aquaponie et l’Internet des Objets (IoT) et l’intelligence artificielle (IA) et le cloud computing. L’ouvrage démarre en exposant les bases conceptuelles essentielles pour comprendre ces technologies ainsi que leurs interactions en se concentrant spécifiquement sur l’aquaponie. Le chapitre consacre ses pages à détailler les éléments fondamentaux de l’aquaponie et ses modes de fonctionnement ainsi que les défis qui surviennent dans la gestion de l’eau et l’écologie. L’étude suit avec l’analyse du rôle croissant de l’Internet des Objets dans l’agriculture de précision ainsi que ses protocoles de communication et applications pratiques. Nous terminons en examinant comment l’intelligence artificielle et le cloud computing interviennent dans la prise de décision et l’administration des systèmes agricoles connectés. Ce résumé expose les possibilités actuelles et leurs barrières pour établir un fondement solide en vue du développement du système SISCA décrit dans les prochains chapitres.

1.2 L’aquaponie : principes et fonctionnemen

1.2.1 Définition de l’aquaponie

L’aquaponie est une nouvelle technologie de production agricole moderne qui combine l’aquaculture et la culture hydroponique. Ainsi, la plantation de légumes ne nécessite plus de fertilisation et les piscicultures ne nécessitent plus de changements d’eau aussi fréquents. Ce changement permet aux poissons, aux cultures et aux micro-organismes de former une symbiose mutuellement bénéfique et une coexistence harmonieuse, garante d’un équilibre écologique. Il s’agit d’un mode de fonctionnement pour une production alimentaire saine et durable [1]. Face à la pollution des sols, à la sécheresse et au changement climatique, les systèmes aquaponiques suscitent un intérêt croissant en raison de leurs économies de ressources, de leur grande efficacité et de leur faible consommation, et sont devenus la tendance et l’orientation du développement agricole moderne [2].

Autrement dit [3] :

$$\text{Aquaponie} = \text{Aquaculture} + \text{Hydroponie}$$

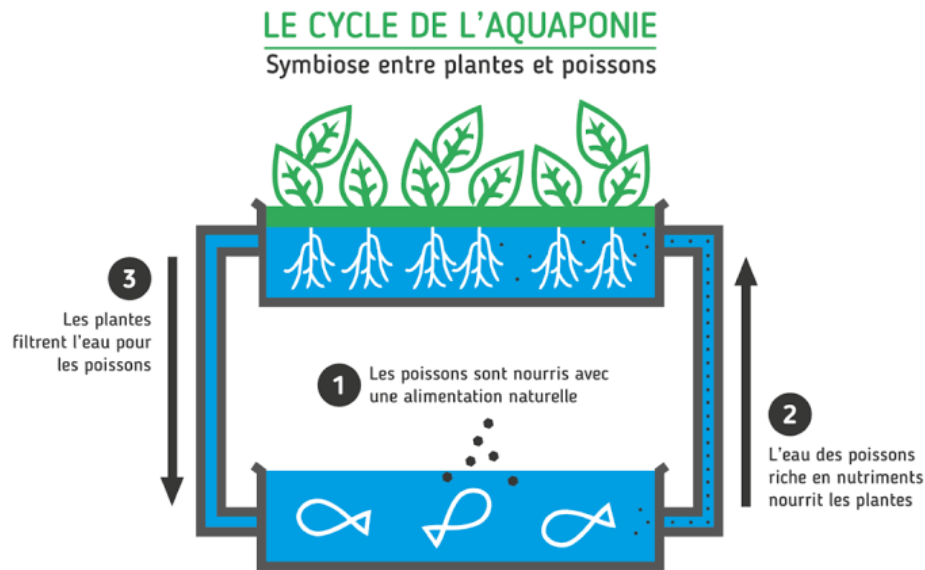


FIGURE 1.1 – Cycle de l'aquaponique [4]

1.2.2 Fonctionnement du système aquaponique

Fonctionnement du système aquaponique comme suit [3] :

- Deux composantes principales :
 - **Aquaculture** : élevage d'organismes aquatiques (ex. : poissons).
 - **Hydroponie** : culture de plantes sans sol.
- Cycle en circuit fermé :
 - L'eau circule entre les deux parties.
 - Les déchets (excréments, nourriture non consommée) s'accumulent dans l'eau.
- Utilisation des effluents :
 - Ces déchets, toxiques à forte concentration pour les poissons, sont riches en nutriments (azote, phosphore, etc.).
 - Les plantes absorbent ces nutriments, ce qui purifie l'eau.
- Sous-systèmes essentiels au bon fonctionnement :
 - Filtration des déchets solides.
 - Régulation du pH par l'ajout de bases.
 - Oxygénation de l'eau pour le bien-être des poissons et des bactéries nitrifiantes.

1.2.3 Avantages de l'aquaponie

- **Économie d'eau** : l'aquaponie utilise jusqu'à 90 % moins d'eau que l'agriculture traditionnelle.
- **Production sans pesticides ni engrais chimiques** : les plantes reçoivent naturellement les nutriments nécessaires à leur croissance.
- **Double production** : possibilité de récolter à la fois des légumes et du poisson.

- **Adaptabilité** : système adapté aux environnements urbains, aux zones arides et aux espaces restreints.
- **Écologique** : réduction des déchets et empreinte carbone minimale.

1.2.4 Défis et considérations

Comme toute technologie, l'aquaponie présente non seulement des avantages, mais également certaines faiblesses qu'il convient de considérer. Selon le rapport de l'Organisation des Nations Unies pour l'alimentation et l'agriculture (FAO), plusieurs inconvénients peuvent freiner son adoption à grande échelle [5] :

1. **Des coûts d'investissement initiaux élevés** : Comparativement à l'hydroponie ou à la culture traditionnelle en sol, la mise en place d'un système aquaponique nécessite un investissement de départ important, notamment en équipements, infrastructure et automatisation.
2. **Une exigence de compétences multidisciplinaires** : La réussite d'un système aquaponique repose sur une connaissance approfondie à la fois des plantes, des poissons et des bactéries. L'opérateur doit également posséder des compétences techniques en plomberie, électricité et gestion de systèmes.
3. **Des besoins biologiques parfois difficiles à concilier** : Il est souvent complexe d'harmoniser les conditions optimales de développement des plantes et des poissons (pH, température, type de substrat, etc.), ce qui peut limiter la diversité des espèces compatibles.
4. **Une flexibilité de gestion réduite** : Comparé aux systèmes autonomes d'aquaculture ou d'hydroponie, l'aquaponie offre moins d'options de gestion, car chaque composant est interdépendant.
5. **Une vulnérabilité élevée aux erreurs** : Une mauvaise gestion du système peut entraîner un effondrement rapide de l'ensemble de l'écosystème, les éléments étant étroitement liés.
6. **Un besoin de suivi quotidien rigoureux** : La gestion quotidienne est incontournable pour garantir l'équilibre du système, ce qui exige une organisation méthodique et une surveillance constante.
7. **Une consommation énergétique non négligeable** : Le système nécessite une alimentation électrique continue pour assurer la circulation de l'eau, l'oxygénation et le fonctionnement des capteurs, générant ainsi des coûts énergétiques récurrents.

1.2.5 Qualité de l'eau en aquaponie

La qualité de l'eau constitue un élément fondamental en aquaponie, car elle influence directement la santé des poissons ainsi que la croissance des plantes. En assurant des conditions optimales, il est possible de favoriser un développement harmonieux et une productivité maximale. À l'inverse, une eau de mauvaise qualité peut provoquer du stress, des maladies, voire la mortalité chez les poissons, tout en freinant la croissance et le rendement des végétaux. Il est donc primordial de maintenir des paramètres de qualité de l'eau adéquats pour garantir le bon fonctionnement du système aquaponique. Pour ce faire, il est essentiel de surveiller et d'analyser régulièrement l'eau du système. Cela implique

la mesure de paramètres clés tels que le pH, les concentrations en ammoniac, en nitrates et en oxygène dissous. Un suivi rigoureux permet de détecter rapidement d'éventuels déséquilibres et d'y remédier avant qu'ils n'affectent l'ensemble du système. Par ailleurs, l'adoption de solutions de filtration et de traitement adaptées contribue à préserver la qualité de l'eau en limitant l'accumulation de substances nocives. En somme, une gestion préventive et rigoureuse de la qualité de l'eau est indispensable à la pérennité et à l'efficacité d'un système aquaponique [6].

On distingue généralement deux grandes catégories de paramètres :

TABLEAU 1.1 – Paramètres qualité de l'eau pour l'aquaculture et pour l'hydroponie.

Paramètres pour l'aquaculture (poissons)	Paramètres pour l'hydroponie (plantes)
Oxygène dissous (DO)	Conductivité électrique (EC)
Température de l'eau	Température de l'air
Ammoniac (NH_3 / NH_4^+)	Humidité de l'air
Nitrites (NO_2^-)	
Nitrites (NO_3^-)	
Turbidité de l'eau	
pH de l'eau	pH de l'eau

1.2.5.1 Paramètres essentiels

Le tableau 1.2 représente les paramètres essentiels de l'eau et son rôle en aquaponie.

TABLEAU 1.2 – Paramètres essentiels de l'eau en aquaponie.

Paramètres	Valeur optimale/ seuil critique	Rôle / Impact
pH	6,8 - 7,2 [6]	Fournit les conditions idéales pour l'absorption des nutriments par les plantes et favorise la croissance des bactéries bénéfiques [6].
Température	22 - 29° C (poissons), 25 - 30° C (bactéries) [7]	La température affecte le métabolisme des poissons et des plantes [7].
Oxygène dissous	≥ 5 mg/L [7]	Les poissons ont besoin d'eau oxygénée pour respirer, tandis que les bactéries bénéfiques ont besoin d'oxygène pour se développer et terminer le processus de nitrification [7].
Ammoniac (NH_3)	0,25 - 2,0 mg/L (forme non ionisée) [8]	Toxique pour les poissons ; forme dépend du pH et température [8].
Nitrites (NO_2^-)	$< 0,25$ mg/L idéale- ment, max 1 mg/L [7]	Très toxique pour les poissons (affecte l'hémoglobine) [7].

Nitrites (NO_3^-)	< 100 - 150 mg/L [8]	Les plantes hydroponiques préfèrent le nitrate comme source d'azote pour leur croissance dans un système aquaponique et l'absorberont par leur système racinaire [8].
CE (conductivité)	1,0 - 2,0 mS/cm [7]	L'EC mesure la capacité de l'eau à conduire l'électricité, qui est corrélée à la concentration de sels dissous (nutriments) dans l'eau [7].
Turbidité	< 5 NTU (unités néphélométriques de turbidité)	Indique la clarté de l'eau; une turbidité élevée peut réduire la lumière, affectant la photosynthèse des plantes et le comportement des poissons.

1.2.5.2 Effets des dépassements des paramètres de qualité de l'eau

TABLEAU 1.3 – Effets des dépassements des paramètres de qualité de l'eau.

Paramètre	Effets en cas de dépassement
pH (< 6 ou > 7,5)	Carences en nutriments (Fe, Zn), inhibition de la nitrification, stress des poissons.
Température (> 30°C ou < 18°C)	Relentissement de la nitrification, stress thermique, baisse de solubilité de l'oxygène.
Oxygène dissous (< 5 mg/L)	Hypoxie, mortalité des poissons, activité bactérienne réduite.
Ammoniac total (> 2 mg/L)	Toxicité aigue pour les poissons, lésions branchiales, arrêt de croissance.
Nitrites (> 0,5 – 1 mg/L)	Intoxication sanguine (formation de méthémoglobine), gaspillage en surface, mortalité.
Nitrites (> 150 mg/L)	Stress osmotique chronique, troubles physiologiques à long terme chez certains poissons.
CE (> 2,5 mS/cm)	Stress osmotique des plantes, brûlures foliaires, absorption réduite d'eau et nutriments.
Turbidité < 5 NTU	Réduction de la pénétration de la lumière; inhibant la photosynthèse des plantes, stress chez les poissons dû à une visibilité réduite; accumulation de matières organiques pouvant favoriser la croissance d'algues nuisibles.

1.2.5.3 Les causes d'une mauvaise qualité d'eau en aquaponie

Plusieurs facteurs peuvent perturber la qualité de l'eau dans un système aquaponique. C'est pourquoi il est essentiel de surveiller et tester régulièrement l'eau. Une détection précoce permet de gérer rapidement tout déséquilibre, tandis qu'une détection tardive peut engendrer de graves problèmes pou-

vant mener à la défaillance du système entier.

Parmi les principales causes d'une mauvaise qualité d'eau en aquaponie figurent [9] :

- **Panne d'équipement** : Des dysfonctionnements inattendus ou des coupures d'électricité peuvent perturber le bon fonctionnement du système.
- **Aération insuffisante** : Un manque d'oxygénation entraîne une diminution de l'oxygène dissous, menaçant la survie des poissons et des plantes.
- **Surpopulation des poissons** : Un nombre excessif de poissons peut provoquer des déséquilibres chimiques dans tout le système.
- **Suralimentation des poissons** : Donner trop de nourriture aux poissons sans éliminer les aliments non consommés entraîne une accumulation de résidus biologiques et un déséquilibre du système.
- **Variations environnementales** : Les changements soudains des conditions environnementales, tels que les variations climatiques et thermiques, impactent fortement la santé du système. Les plantes et les poissons sont particulièrement sensibles aux fluctuations brusques de température ou aux conditions météorologiques extrêmes, comme le gel ou la chaleur intense.
- **Déséquilibres chimiques** : L'usage d'additifs chimiques tels que les fertilisants peut gravement perturber l'équilibre du système aquaponique.
- **Mauvais choix des poissons/plantes** : Choisir des espèces de poissons et de plantes incompatibles avec le système choisi peut entraîner des dysfonctionnements importants ou nécessiter une remise à zéro complète du système. Il est essentiel de respecter le rapport approprié entre poissons et plantes, et de sélectionner soigneusement les espèces adaptées au type de système aquaponique envisagé.
- **Stress des poissons** : La santé des poissons est cruciale pour le bon fonctionnement d'un système aquaponique. Maintenir vos poissons dans un environnement sain et adapté est donc indispensable à la réussite globale du système.

1.3 L'Internet des Objets (IoT)

1.3.1 Définition de l'IoT dans l'agriculture

L'Internet des objets (IoT, pour Internet of Things en anglais) fait référence à un groupe d'objets matériels liés à différents réseaux et technologies de logiciels, comme des capteurs, appareils électroniques et sondes. Ils sont interconnectés par Internet afin de transmettre et/ou recevoir des données. L'objectif est bien de centraliser l'information, d'enrichir aussi les processus décisionnels et d'assurer en plus une communication fluide entre les composantes du système [10].

Puisque l'IoT automatise la collecte de données tout comme il réduit l'intervention humaine, alors il facilite l'accès à l'information pertinente de manière considérable. Ce potentiel rend cette technologie particulièrement adaptée pour le domaine agricole, où réactivité en temps réel et supervision intelligente sont des enjeux majeurs. la figure 1.2 explique plus en détail le concept de l'IoT.

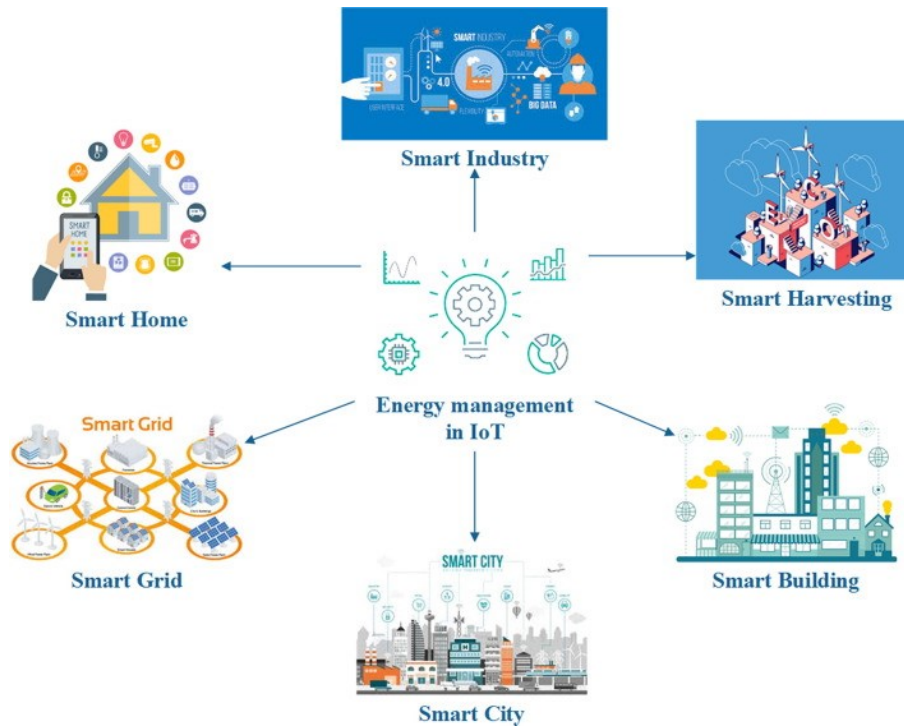


FIGURE 1.2 – Image description pour l'IoT [11].

1.3.2 Caractéristiques de l'IoT

L'Internet des Objets (IoT) se distingue par plusieurs caractéristiques clés qui le différencient des systèmes informatiques traditionnels. Voici les principales caractéristiques de l'IoT [12] :

- **Interconnectivité :**

Dans le contexte de l'Internet des Objets, tout objet peut être relié à l'infrastructure mondiale d'information et de communication.

- **Services liés aux objets :**

L'IoT permet de fournir des services centrés sur les objets, tels que la protection de la vie privée et la cohérence sémantique entre les objets physiques et leurs équivalents virtuels. Pour assurer ces services, tant les aspects physiques que les technologies de l'information doivent évoluer pour respecter les contraintes propres aux objets.

- **Hétérogénéité :**

Les dispositifs IoT sont hétérogènes, car ils reposent sur une grande diversité de plateformes matérielles et de réseaux. Ils sont capables de communiquer entre eux, même lorsqu'ils utilisent des réseaux différents.

- **Évolution dynamique :**

L'état des appareils IoT évolue en permanence : ils peuvent être en veille ou actifs, connectés ou déconnectés, et leur contexte (comme la localisation ou la vitesse) peut aussi changer. De plus, le nombre d'appareils connectés peut varier dynamiquement.

- **Échelle massive :**

Le nombre d'appareils à gérer et à faire interagir dans l'IoT sera au moins dix fois supérieur à

celui des dispositifs actuellement connectés à Internet. La gestion et l'interprétation de ces objets, notamment au niveau sémantique, représenteront un enjeu majeur.

- **Sécurité :**

La sécurité doit rester une priorité tout en tirant parti des avantages de l'IoT. Il est essentiel de concevoir des systèmes sûrs, tant pour les fabricants que pour les utilisateurs. Cela implique de garantir la sécurité des données, des réseaux, des points d'accès physiques, ainsi que des informations qui y circulent, nécessitant ainsi le développement d'un modèle de sécurité évolutif.

- **Connectivité :**

La connectivité est indispensable pour assurer la compatibilité avec l'accès réseau. L'accès correspond à la capacité de se connecter à un réseau, tandis que la compatibilité désigne la capacité partagée de consommer et de générer des données.

1.3.3 Avantages de l'IoT en agriculture

- **Optimisation des ressources :** consommation de l'eau, des engrais et des pesticides réduits.
- **Améliorer les rendements :** grâce à une meilleure compréhension des besoins des cultures.
- **Réduisez les coûts :** automatisez alors les tâches puis diminuez les pertes.
- **Traçabilité :** suivi précis à l'égard des produits.

1.3.4 Applications de l'IoT en agriculture de précision

L'IoT offre de nombreuses applications dans le domaine agricole, notamment :

- **Surveillance des conditions climatiques :** Les stations météorologiques intelligentes, intégrées dans des systèmes IoT, collectent des données environnementales en temps réel (température, humidité, précipitations, etc.), permettant aux agriculteurs d'ajuster leurs pratiques agricoles en fonction des conditions météorologiques [13].
- **Gestion de l'irrigation :** Des capteurs d'humidité du sol permettent d'optimiser l'irrigation, réduisant ainsi la consommation d'eau et améliorant la santé des cultures. Des capteurs d'humidité du sol, connectés via des réseaux IoT, permettent d'optimiser l'irrigation en fournissant des données précises sur la teneur en eau du sol. Cela réduit la consommation d'eau et améliore la santé des cultures [14].
- **Suivi de la santé des cultures :** L'utilisation de capteurs et de drones équipés de caméras multispectrales permet de détecter les signes de stress ou de maladie des plantes, facilitant une intervention rapide et ciblée [15].
- **Automatisation des équipements agricoles :** Les tracteurs et autres machines agricoles peuvent être automatisés et contrôlés à distance grâce à l'IoT, améliorant ainsi l'efficacité des opérations et réduisant les coûts de main-d'œuvre [13].

1.3.5 Défis futurs de l'Internet des Objets (IoT)

L'Internet des Objets (IoT) ne cesse de croître et d'évoluer avec le temps, ouvrant la voie à de nouvelles opportunités, mais aussi à de nombreux défis. Voici quelques-uns des principaux défis auxquels

l'IoT pourrait être confronté :

- **Vie privée et sécurité :**

Avec l'IoT qui devient un élément central de l'Internet du futur, notamment dans des systèmes critiques à grande échelle, la question de la confiance et de la sécurité devient primordiale. Les nouveaux défis identifiés en matière de vie privée, de confiance et de fiabilité incluent :

- Garantir la fiabilité et la qualité des informations dans des modèles de données partagées afin de permettre leur réutilisation dans de multiples applications.
- Assurer un échange de données sécurisé entre les objets connectés et les utilisateurs ou consommateurs de ces informations.

- **Coût et facilité d'utilisation :**

L'IoT connecte des objets physiques à Internet grâce à diverses technologies. Pour favoriser une adoption massive dans les années à venir, le coût des composants nécessaires pour activer des fonctions telles que la détection, le suivi ou le contrôle doit rester suffisamment bas afin de garantir l'accessibilité et la rentabilité de ces systèmes.

- **Interopérabilité :**

L'interopérabilité est une propriété fondamentale de l'Internet traditionnel : pour être « connectés », les appareils doivent « parler le même langage » en termes de protocoles et de formats de données. Actuellement, plusieurs secteurs utilisent différents standards selon leurs besoins. Face à la multiplicité des sources de données et à l'hétérogénéité des dispositifs, il devient essentiel de proposer des interfaces cohérentes et compatibles, surtout pour les applications impliquant plusieurs organisations ou systèmes. L'IoT devra donc garantir un haut niveau d'interopérabilité.

- **Gestion des données :**

La gestion des données représente un enjeu majeur pour l'IoT. Dans un monde où des objets en réseau échangent en permanence toutes sortes d'informations, la volumétrie des données générées et les méthodes de traitement associées prennent une importance considérable.

- **Problèmes d'alimentation au niveau des dispositifs :**

Trouver des solutions pour connecter les objets de manière interopérable tout en tenant compte des contraintes énergétiques constitue un défi majeur. En effet, la connectivité reste souvent le processus le plus énergivore pour les dispositifs IoT.

1.3.6 Protocoles de l'IoT

La valeur et l'utilité de l'Internet des objets (IoT) résident dans la capacité des composants à communiquer entre eux. Cette communication s'effectue grâce à des protocoles IoT, qui garantissent que les données envoyées par les dispositifs terminaux, comme les capteurs, sont correctement reçues et comprises par les étapes suivantes de l'environnement connecté, que ce soit un autre appareil, une passerelle ou une application. En d'autres termes, les protocoles IoT sont aussi essentiels à l'existence de l'IoT que les objets eux-mêmes. Même si les protocoles, pris dans leur ensemble, sont indispensables au bon fonctionnement de l'IoT, ils ne sont pas tous équivalents. En effet, tous les protocoles ne fonctionnent pas – ou ne fonctionnent pas efficacement – dans toutes les situations, comme l'explique

Bill Ray, analyste et directeur de recherche principal chez Gartner.

Selon Ray, certains protocoles sont bien adaptés aux usages IoT à l'intérieur des bâtiments, d'autres conviennent mieux aux déploiements répartis entre plusieurs bâtiments, tandis que d'autres encore sont plus appropriés pour des cas d'usage IoT à l'échelle nationale ou mondiale [16].

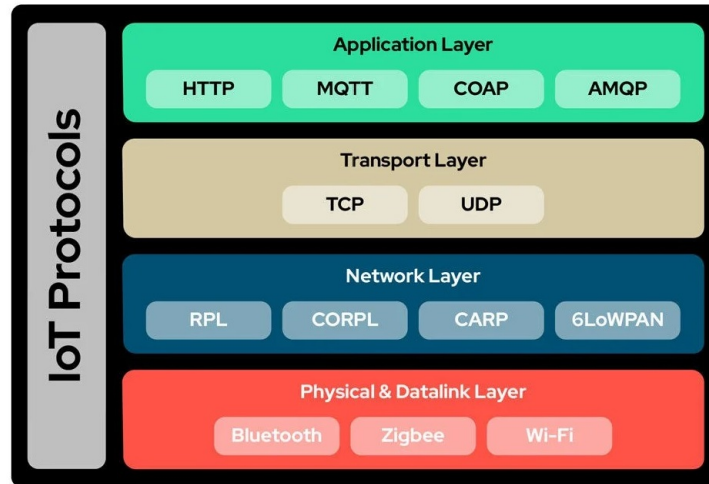


FIGURE 1.3 – Protocoles IoT [17].

1.3.7 Protocoles les plus courants

Les spécialistes en technologie peuvent choisir parmi plusieurs protocoles de communication lors de la création d'un réseau destiné à leur écosystème IoT.

1.3.7.1 Protocole de Communication – MQTT

Le protocole MQTT (Message Queuing Telemetry Transport) repose sur l'architecture TCP/IP et suit un modèle de communication de type "publish-subscribe".

MQTT est un protocole orienté événements. Pour limiter le volume de transmissions, les données ne sont ni envoyées en continu ni à intervalles réguliers. Un client publie uniquement lorsqu'il dispose de nouvelles informations, et le broker ne transmet ces données aux abonnés que lorsqu'il en reçoit [18].

Mode de fonctionnement

Dans cette architecture, on distingue deux entités principales : les clients et le broker (ou courtier). Le broker joue le rôle de serveur central chargé de recevoir les messages des clients et de les redistribuer aux autres clients abonnés. Ainsi, les échanges ne se font jamais directement entre clients, mais toujours par l'intermédiaire du broker. Un client peut agir en tant qu'éditeur, abonné, ou assumer ces deux rôles simultanément [18].

L'image 1.4 ci-dessous montre comment fonctionne le protocole MQTT. Les capteurs (à gauche) envoient des données, comme la température, au serveur central appelé broker. Ce broker reçoit les messages et les transmet aux appareils abonnés (à droite), comme les ordinateurs. Les capteurs publient les informations, et les ordinateurs les reçoivent s'ils sont abonnés au bon sujet. Les capteurs et les ordinateurs ne communiquent pas directement entre eux, tout passe par le broker.

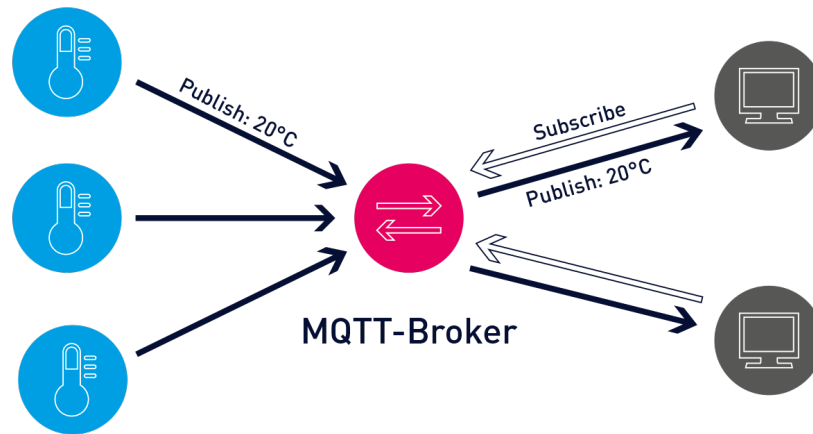


FIGURE 1.4 – Exemples de topologies MQTT [18].

Les avantages du protocole MQTT

Le protocole MQTT est largement utilisé dans l'IoT en raison de ses nombreux avantages [19] :

- **Léger et efficace** : Idéal pour les microcontrôleurs grâce à sa faible consommation de bande passante et ses messages de petite taille.
- **Évolutif** : Peut gérer des millions de dispositifs avec un code minimal et une faible consommation d'énergie.
- **Fiable** : Fonctionne bien sur des réseaux instables avec trois niveaux de qualité de service pour assurer la livraison des messages.
- **Sécurisé** : Prend en charge le chiffrement et l'authentification via TLS, OAuth, et certificats.
- **Facile à intégrer** : Compatible avec de nombreux langages comme Python, facilitant son déploiement dans divers projets.

1.4 Intelligence Artificielle (IA)

1.4.1 Définition de l'intelligence artificielle (IA)

L'**intelligence artificielle** (IA) est une branche de l'informatique qui simule les processus cognitifs humains. Elle inclut des techniques telles que l'apprentissage automatique, l'apprentissage profond et les réseaux neuronaux. L'objectif ultime de l'IA est de créer des systèmes capables d'analyser des données complexes. Ces systèmes **apprennent** de leurs analyses et **prennent des décisions** de manière autonome.

Dans le domaine de l'agriculture, l'IA permet de transformer la manière dont les cultures sont surveillées, nourries et récoltées. En utilisant des capteurs et des algorithmes d'apprentissage automatique, les agriculteurs obtiennent des informations précises sur la santé de leurs cultures. Ils peuvent également suivre les besoins en eau, en nutriments et bien plus encore, en temps réel [20] :

- Analyse des données complexes.
- Prise de décisions autonome.
- Amélioration de l'efficacité agricole.

1.4.2 Avantages de l'IA en agriculture

Les avantages de l'IA en agriculture sont [21] :

- **Efficacité et productivité :**

- L'IA optimise les opérations agricoles (semis, irrigation, récolte) grâce à l'analyse de données.
- Capteurs et drones réduisent le temps et l'effort nécessaires à la surveillance des cultures.
- Les algorithmes prédictifs aident à anticiper les besoins des plantes, améliorant rendement et qualité.

- **Réduction des coûts :**

- L'automatisation (tracteurs autonomes, pulvérisation de précision) diminue les besoins en main-d'œuvre.
- Optimisation de l'usage de l'eau, des engrais et des pesticides, limitant les dépenses.
- Moins de gaspillage et meilleure gestion des ressources.

- **Durabilité environnementale :**

- L'IA permet un usage raisonné des intrants chimiques, réduisant l'impact écologique.
- L'agriculture de précision aide à préserver les sols, l'eau et la biodiversité.
- Les décisions en temps réel améliorent la gestion durable des cultures face au climat.

1.4.3 Limites de l'IA en agriculture

Les limites de l'IA en agriculture sont [22] :

- **Précision des données limitée :**

Si les données collectées sont biaisées ou incomplètes, les décisions prises par l'IA peuvent être inexactes ou contre-productives.

- **Risque de cyberattaques :**

Les systèmes d'IA peuvent être la cible de piratages, mettant en danger les données sensibles et le bon fonctionnement des exploitations agricoles.

- **Usage non éthique possible :**

L'IA peut être détournée pour manipuler le marché, fausser la production ou fixer des prix injustes.

- **Dépendance technologique :**

Une trop grande automatisation peut réduire l'expertise humaine et créer une dépendance aux systèmes numériques.

- **Accès inégal aux technologies :**

Les petits agriculteurs peuvent ne pas avoir les moyens d'adopter l'IA, creusant ainsi les inégalités dans le secteur agricole.

1.4.4 L'avenir de l'IA en agriculture

Pour nourrir les 09,2 milliards d'habitants prévus d'ici 2050, l'agriculture devra devenir plus intelligente et performante, car les terres cultivables n'augmenteront que de 04 %. Il ne s'agit donc pas de cultiver plus, mais de cultiver mieux.

L'intelligence artificielle jouera un rôle central dans cette transformation en relevant quatre défis majeurs [22] :

- **Prévision** : améliorer l'anticipation des conditions climatiques pour optimiser les rendements.
- **Compétitivité** : augmenter la productivité tout en réduisant les coûts d'exploitation.
- **Durabilité** : limiter l'usage des ressources (eau, énergie) et protéger l'environnement.
- **Gain de temps** : automatiser les tâches répétitives et améliorer les revenus des agriculteurs.

1.5 Cloud computing

Aujourd'hui, nous vivons à l'ère de l'informatique en nuage (« cloud computing »), une technologie qui occupe une place centrale dans le domaine des technologies de l'information. Basé sur Internet, le cloud computing représente l'architecture informatique la plus puissante, réunissant un ensemble de matériels, de logiciels et d'infrastructures interconnectés à travers le réseau.

Au-delà du simple traitement en réseau et des calculs distribués, le cloud offre de nombreux avantages supplémentaires. Dans ce qui suit, nous présentons les notions fondamentales liées au cloud computing, notamment sa définition, ses principales caractéristiques, ainsi que ses différents types et modèles de services.

1.5.1 Définition Cloud computing

L'informatique en nuage (cloud computing) désigne le fait de stocker et d'accéder à des données et des applications via Internet, plutôt que sur le disque dur local d'un ordinateur. Sur les schémas de réseaux informatiques, Internet est d'ailleurs souvent représenté sous la forme d'un nuage (« cloud »). Le cloud computing consiste à exploiter des ressources matérielles et logicielles pour fournir des services à travers un réseau, le plus souvent Internet. Cela permet aux utilisateurs d'accéder à leurs données et à leurs applications depuis n'importe quel ordinateur ou appareil disposant d'une connexion Internet. Un exemple courant de service de cloud computing est la messagerie Gmail de Google [23].

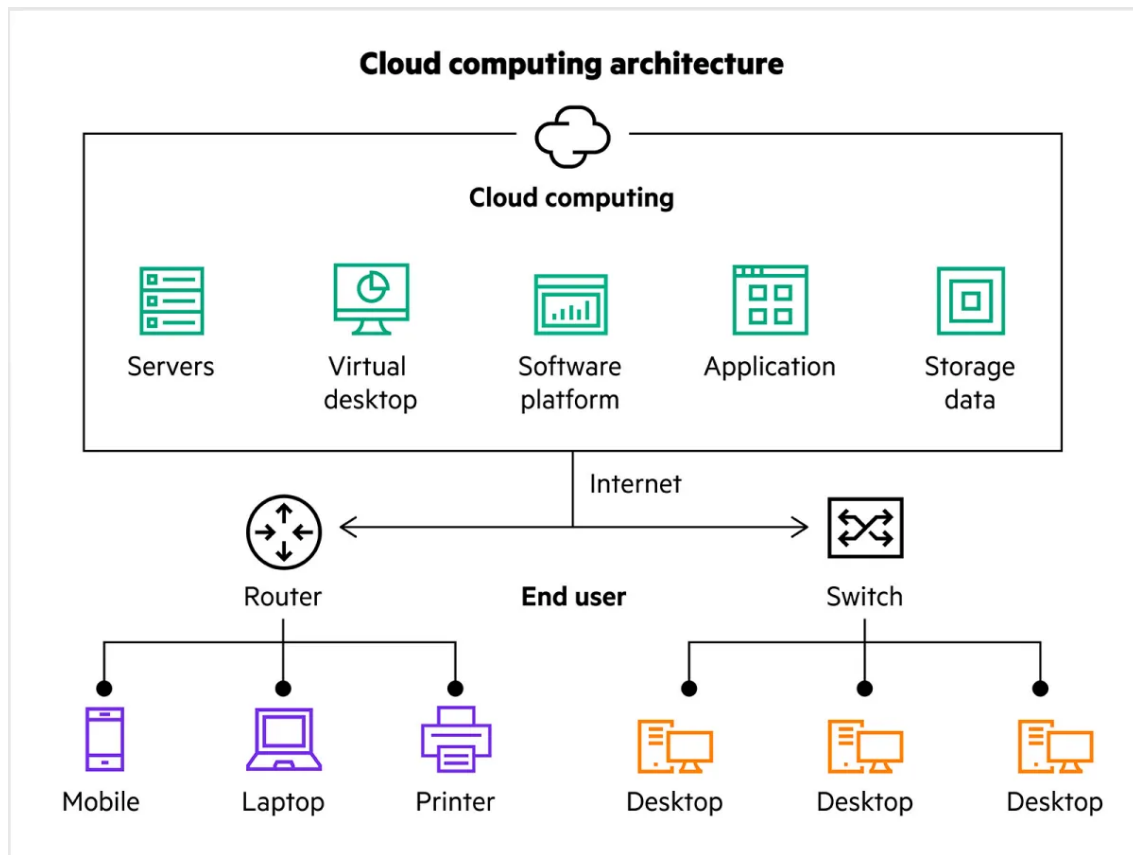


FIGURE 1.5 – Présentation du Cloud Computing [24].

Le National Institute of Standards and Technology (NIST) propose une définition largement acceptée du cloud computing :

« Le cloud computing est un modèle permettant un accès réseau omniprésent, pratique et à la demande à un pool partagé de ressources informatiques configurables (par exemple, réseaux, serveurs, stockage, applications et services) pouvant être rapidement provisionnées et mises à disposition avec un minimum d'efforts de gestion ou d'interaction avec le fournisseur de services. Ce modèle de cloud favorise la disponibilité et se compose de cinq caractéristiques essentielles, de trois modèles de service et de quatre modèles de déploiement. »

1.5.2 Caractéristiques du Cloud Computing

Le cloud computing présente plusieurs caractéristiques qui en font une technologie prisée et efficace pour les entreprises et organisations. Parmi les principales caractéristiques, on retrouve [25] :

- **Accès étendu au réseau (mobilité) :**

Les services de cloud computing sont accessibles via Internet ou d'autres réseaux étendus, à partir de n'importe quel appareil disposant d'une connexion internet (smartphone, ordinateur, etc.), et depuis n'importe quel endroit, assurant ainsi une accessibilité ubiquitaire.

- **Grande scalabilité :**

L'infrastructure du cloud est hautement évolutive. Les fournisseurs de services cloud peuvent

ajouter de nouveaux serveurs et nœuds avec peu de modifications sur l'infrastructure et les logiciels existants.

- **Haute disponibilité et fiabilité :**

Grâce à la redondance des serveurs et à leur grande disponibilité, les risques de défaillance de l'infrastructure sont considérablement réduits.

- **Maintenance simplifiée :**

Les applications cloud sont accessibles depuis différents emplacements et n'ont pas besoin d'être installées sur chaque machine, ce qui simplifie la maintenance et réduit les coûts associés.

- **Services mesurés :**

Les ressources du cloud sont généralement facturées à l'utilisation. Ainsi, les organisations ne paient que pour les ressources qu'elles consomment réellement.

- **Élasticité rapide :**

Les ressources du cloud peuvent être augmentées ou réduites rapidement en fonction des besoins, offrant ainsi aux entreprises la flexibilité nécessaire pour s'adapter à la demande sans investir dans leur propre matériel.

- **Rentabilité :**

Les services proposés par les fournisseurs de cloud sont très abordables, voire parfois gratuits. Le modèle de facturation à la consommation (« pay as you go ») élimine la nécessité d'acheter une infrastructure, ce qui réduit les coûts de maintenance.

- **Libre-service à la demande :**

Le cloud permet aux utilisateurs de provisionner rapidement et facilement les ressources informatiques nécessaires, sans intervention humaine de la part du fournisseur de services.

- **Mutualisation des ressources :**

Les ressources du cloud (puissance de calcul, stockage, bande passante, etc.) sont partagées entre plusieurs utilisateurs afin d'optimiser leur utilisation et leur efficacité.

- **Multi-partage :**

Un cloud offre des services à plusieurs utilisateurs simultanément. Chaque utilisateur bénéficie d'un environnement virtuel isolé tout en partageant les ressources réseau, hôte et applicatives du cloud.

- **Virtualisation :**

Le cloud permet aux utilisateurs d'accéder aux services depuis n'importe quel terminal, où qu'ils soient. Les ressources proviennent du cloud plutôt que d'entités physiques visibles. Grâce à un ordinateur portable ou un smartphone connecté, les utilisateurs peuvent accomplir diverses tâches à distance, de manière sécurisée et à tout moment. Ainsi, des tâches trop complexes pour un seul ordinateur peuvent être réalisées grâce à la puissance collective du cloud.

1.5.3 Les types de Cloud Computing

Les types de Cloud Computing sont [26] :

- **Cloud public :**

Le cloud public est accessible à tous les clients externes via Internet. Ces utilisateurs peuvent s'inscrire auprès du fournisseur de cloud et utiliser les ressources proposées selon un modèle de paiement à l'usage (« pay-per-use »). Contrairement au cloud privé, le cloud public est moins sécurisé en raison de son ouverture à tous les internautes. Il offre moins de possibilités de personnalisation et est géré par un grand fournisseur de services cloud (CSP), qui en assure la maintenance et la disponibilité. Le cloud public, parfois appelé « cloud externe » ou « cloud ouvert », met ainsi ses ressources à disposition de manière mutualisée pour un large public.

- **Cloud privé :**

Le cloud privé est déployé spécifiquement pour une entreprise ou une organisation, généralement au sein de son propre centre de données. L'organisation garde le contrôle total sur ses ressources cloud et en assure la gestion. Comparé au cloud public ou hybride, le cloud privé garantit un niveau de sécurité supérieur et une meilleure maîtrise des données. Même si le coût du cloud privé est généralement plus élevé que celui du cloud public, il offre des performances et une productivité accrues pour les utilisateurs concernés. Le cloud privé, également appelé « cloud interne » ou « cloud d'entreprise », permet de restreindre l'accès aux seuls membres de l'organisation derrière un pare-feu.

- **Cloud hybride :**

Le cloud hybride combine des éléments de clouds publics et privés, ainsi que, parfois, de clouds communautaires. Les tâches critiques ou sensibles sont généralement traitées dans le cloud privé, tandis que les activités moins stratégiques ou nécessitant moins de sécurité sont déléguées au cloud public. Ce modèle permet d'optimiser les coûts tout en bénéficiant d'une flexibilité accrue. La gestion d'un cloud hybride nécessite une coordination entre l'infrastructure interne et les services externes, afin d'assurer la redondance et la disponibilité des données. Par exemple, une entreprise peut stocker des données sensibles sur un cloud privé tout en utilisant le cloud public pour des applications moins critiques.

- **Cloud communautaire :**

Le cloud communautaire est mis en place et partagé par plusieurs organisations ayant des besoins, des politiques ou des objectifs communs. L'infrastructure cloud est gérée par un groupe restreint d'entreprises ou éventuellement par un tiers, et sert une communauté d'utilisateurs partageant les mêmes intérêts. Ce modèle offre un bon équilibre entre sécurité, partage des ressources et maîtrise des coûts. Les clouds communautaires sont généralement plus sûrs que les clouds publics et favorisent la collaboration entre les membres de la communauté concernée.

1.5.4 Modèles de services du Cloud Computing

Les services cloud permettent au cloud d'interagir avec un client (utilisateur ou application) de différentes manières. On distingue trois principaux modèles de services :

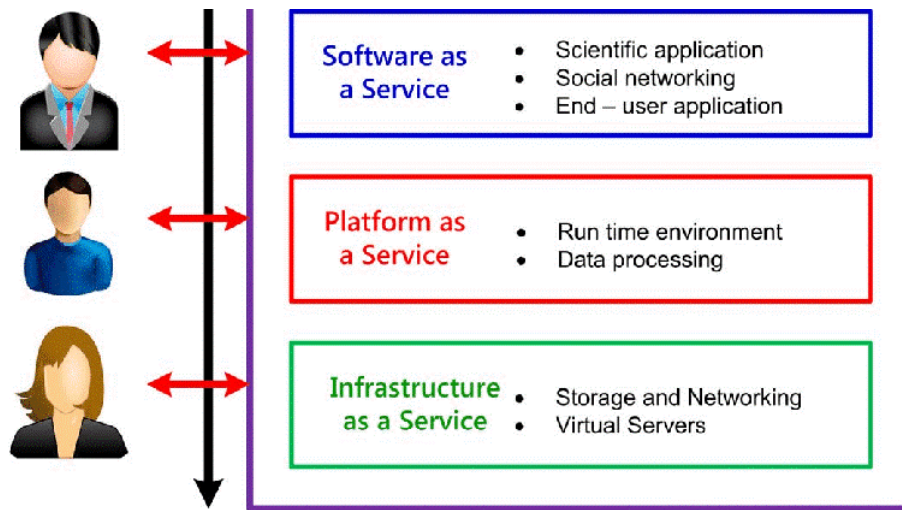


FIGURE 1.6 – Modèle de Cloud Computing [27].

1.5.4.1 SaaS – Software as a Service

- Définitions

Le Software as a Service, ou SaaS, est un modèle de service basé sur le cloud dans lequel les utilisateurs accèdent à des applications via un navigateur web, plutôt que de devoir les télécharger et les installer sur un ordinateur ou un réseau local. Ce type de service peut inclure une grande variété d'applications professionnelles, comme des outils de bureautique ou de communication unifiée [28].

- Les avantages du SaaS

Les avantages du SaaS sont [28] :

• Accessibilité :

L'un des grands atouts du SaaS est sa compatibilité universelle : accessible via un simple navigateur, peu importe le système d'exploitation (Windows, macOS, Linux, Android ou iOS). Cela permet une utilisation fluide sur ordinateurs et appareils mobiles, même en déplacement.

• Mises à jour :

Les mises à jour sont gérées centralement par le fournisseur, sans interruption pour l'utilisateur. Contrairement aux logiciels traditionnels, il n'est pas nécessaire de tester ou valider les correctifs localement, ce qui garantit une sécurité et une innovation continues.

• Matériel :

Le SaaS élimine le besoin d'investissements lourds en infrastructure informatique. Pas besoin de serveurs ou de configurations complexes : même les petites entreprises peuvent accéder à des outils performants auparavant réservés aux grandes structures. Il est également facilement évolutif selon les besoins.

• Portée du marché :

Pour les éditeurs, le SaaS permet d'élargir leur clientèle sans contrainte technique, tout en rendant les services plus abordables. Pour les utilisateurs, cela signifie un accès à des outils professionnels de pointe, quel que soit leur budget.

- **Stockage et mobilité :**

Les données sont automatiquement sauvegardées dans le cloud, garantissant leur sécurité et leur accessibilité depuis n'importe quel appareil. Les utilisateurs peuvent ainsi reprendre leur travail là où ils l'ont laissé, sans perte de données.

- **Données et analyses :**

Grâce à la centralisation, les données peuvent être facilement exploitées via des outils d'analyse intégrés. Les entreprises bénéficient d'une meilleure visibilité sur leurs opérations, tandis que les fournisseurs réduisent les risques liés au piratage grâce au modèle par abonnement.

- Architecture SaaS

Le système SISCA adopte une architecture **multi-tenant**, permettant à plusieurs utilisateurs d'accéder à une même plateforme SaaS tout en maintenant une séparation stricte de leurs données via des identifiants (`user_id`). Cette approche garantit une meilleure scalabilité, une réduction des coûts de déploiement, et une gestion centralisée plus efficace du système.

L'illustration 1.7 ci-dessous présente la structure typique d'une architecture multi-tenant :

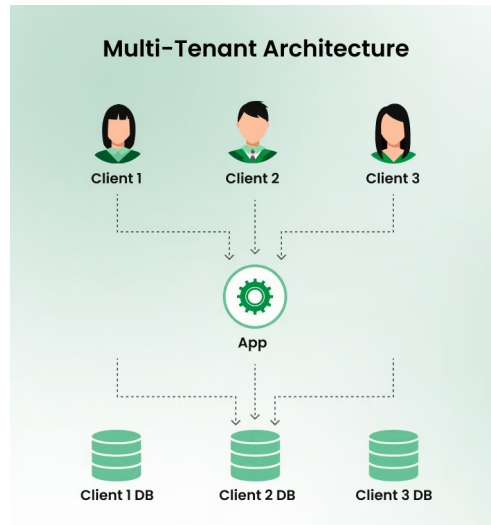


FIGURE 1.7 – Architecture multi-tenant [29].

- Inconvénients du SaaS

- Risque d'interruption des services.
- Nécessité d'une connexion Internet constante et rapide.
- Impossibilité d'utiliser le logiciel en cas de panne système ou de connexion.

1.5.4.2 PaaS – Platform as a Service (Plateforme en tant que Service)

- Définitions

Le PaaS offre des ressources logicielles et informatiques (telles que les systèmes d'exploitation, outils de développement, bases de données, etc.) gérées à distance par le fournisseur cloud. Les services incluent l'intégration de services en ligne, la collaboration, le développement, l'hébergement et la gestion d'applications. Les utilisateurs n'ont pas à se soucier de la gestion matérielle ou logicielle

ni à employer des spécialistes pour cette tâche. Cependant, l'interopérabilité et la portabilité entre fournisseurs restent limitées.

Exemples de PaaS : Microsoft Azure, Google App Engine, Rackspace Cloud Sites, Salesforce.com [30].

- Avantages du PaaS

- Flexibilité accrue du processus de développement et réduction de la gestion du stockage serveur.
- Sécurité renforcée, incluant la sauvegarde et la récupération des données.
- Réduction des coûts grâce à la location des ressources sans nécessité d'infrastructure spécialisée.

- Inconvénients du PaaS

- L'utilisateur n'a aucun contrôle direct sur la machine virtuelle qui gère ses données.
- Principalement adapté aux applications web.
- Des compétences de base en programmation sont nécessaires pour intégrer efficacement le modèle PaaS dans le système.

1.5.4.3 IaaS – Infrastructure as a Service (Infrastructure en tant que Service)

- Définitions

Dans le modèle IaaS, le fournisseur de services cloud propose des ressources informatiques virtualisées telles que processeurs, mémoire, systèmes d'exploitation, logiciels applicatifs, etc. La technologie de virtualisation permet de transformer les ressources physiques en ressources logiques que les clients peuvent allouer et libérer dynamiquement selon leurs besoins.

Exemples de fournisseurs : Rackspace Cloud Servers, Google, Amazon EC2, IBM, Verizon [30].

- Avantages de l'IaaS

- Possibilité d'exploiter des machines virtuelles à la demande grâce à la virtualisation.
- Mise à disposition de services réseau tels que l'équilibrage de charge, les routeurs ou pare-feu.
- Réduction des dépenses en ressources humaines et matérielles.

- Inconvénients de l'IaaS

- Dépendance au réseau et latence éventuelle.
- Complexité technique, car l'IaaS nécessite des compétences spécialisées pour la configuration et la gestion de l'infrastructure.
- Les interruptions de service ou les pannes chez le fournisseur peuvent impacter l'accès aux ressources et les opérations de l'organisation.

1.5.5 Défis de l'intégration du Cloud Computing et de l'IoT

L'intégration du cloud et de l'IoT présente de nombreux avantages et favorise l'émergence ou l'amélioration d'applications innovantes. Toutefois, ce scénario complexe, appelé CloudIoT, pose plusieurs défis majeurs qui attirent l'attention de la communauté scientifique. Voici les principaux défis identifiés [31] :

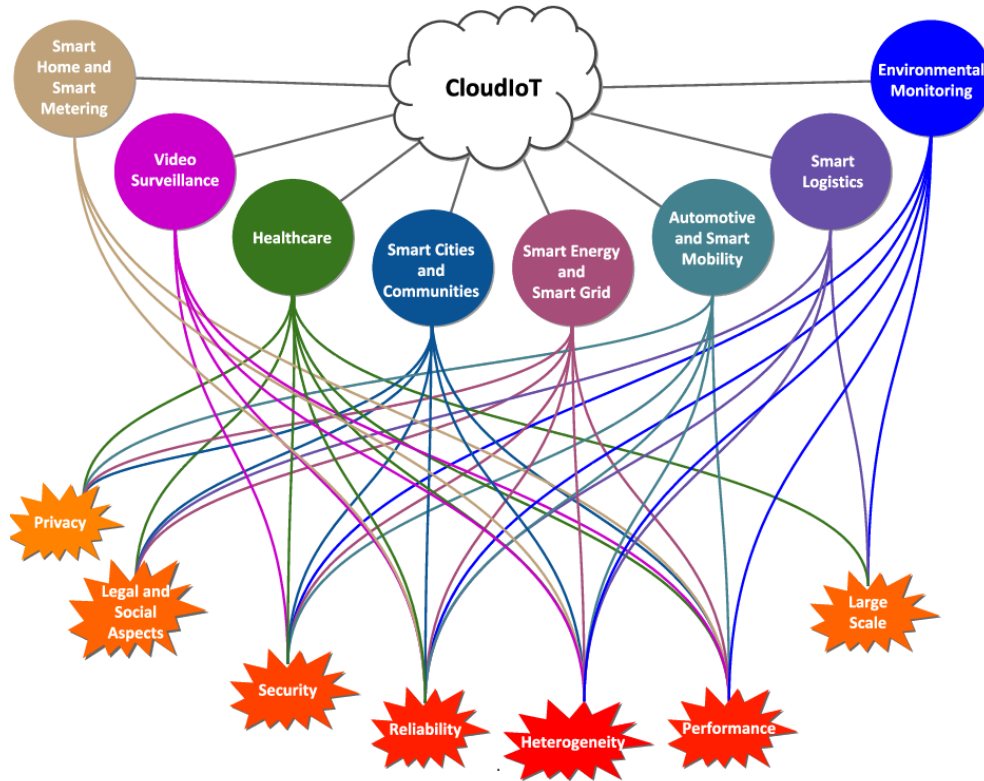


FIGURE 1.8 – Scénarios d’application pilotés par le paradigme CloudIoT et défis associés [31].

- **Sécurité et confidentialité :**

Lorsque des applications IoT critiques sont migrées vers le cloud, des préoccupations apparaissent, notamment en ce qui concerne la confiance envers le fournisseur de services, la compréhension des accords de niveau de service (SLA), ou encore la localisation réelle des données. De nouveaux enjeux nécessitent une attention particulière, car un système distribué comme celui-ci reste vulnérable à de nombreux types d’attaques (par exemple : détournement de session, injection SQL, cross-site scripting, attaques par canal auxiliaire) et à des failles critiques (comme le détournement de session ou l’évasion de machines virtuelles). De plus, la multi-location peut compromettre la sécurité et exposer des informations sensibles. Les limitations de puissance de calcul des objets connectés empêchent parfois l’utilisation de la cryptographie à clé publique à tous les niveaux. Autant de questions sur lesquelles la recherche se penche actuellement pour améliorer la sécurité et la confidentialité dans les architectures CloudIoT.

- **Hétérogénéité :**

L’un des principaux défis de l’intégration cloud-IoT réside dans la grande diversité des dispositifs, des systèmes d’exploitation, des plateformes et des services utilisés pour le développement de nouvelles applications.

- **Performance :**

L’intégration du cloud et de l’IoT s’accompagne souvent d’exigences particulières en termes de performance et de qualité de service (QoS) à différents niveaux (communication, calcul, stockage). Répondre à ces exigences peut s’avérer complexe dans certains cas.

- **Fiabilité :**

Lorsque l'intégration du cloud et de l'IoT concerne des applications critiques, la question de la fiabilité devient primordiale. Par exemple, dans le domaine de la mobilité intelligente, les véhicules sont souvent en mouvement, rendant les communications véhiculaires intermittentes et peu fiables. Dans les environnements à ressources limitées, on observe également des problèmes liés à la défaillance ou à l'indisponibilité des appareils.

- **Big Data :**

Avec l'explosion attendue du nombre d'objets connectés (plus de 50 milliards prévus à l'horizon 2020), la gestion du volume massif de données générées pose des défis considérables en matière de transfert, de stockage, d'accès et de traitement. L'omniprésence des capteurs et des terminaux mobiles nécessite des systèmes informatiques hautement évolutifs.

- **Supervision et monitoring :**

Le monitoring est une activité essentielle dans les environnements cloud, notamment pour la planification des capacités, la gestion des ressources, le respect des SLA, la performance, la sécurité et le diagnostic des incidents. Avec l'intégration de l'IoT, cette tâche devient encore plus critique et complexe.

- **Passage à grande échelle :**

CloudIoT permet le développement d'applications innovantes qui intègrent et analysent des données issues d'objets connectés répartis sur de vastes territoires. L'interaction avec un très grand nombre de dispositifs, souvent disséminés sur de larges zones, accentue la complexité des systèmes. De plus, le déploiement massif d'objets connectés rend le monitoring plus difficile, notamment en raison des problèmes de latence et de connectivité.

1.6 Travaux similaires sur l'aquaponie intelligente

Nemade et Shah (2023) ont développé un système prédictif destiné à améliorer la gestion des systèmes aquaponiques intégrant poissons d'eau chaude et froide, plantes et bactéries. Des capteurs IoT collectent les données en temps réel, suivies par un nettoyage approfondi de ces données et une corrélation des attributs pour éliminer ceux qui ne sont pas pertinents. Pour résoudre le problème classique du déséquilibre des classes, les auteurs ont utilisé des classificateurs de machine learning multiples avec un mécanisme de vote pour déterminer les prédictions finales. Contrairement à notre étude, qui utilise des techniques avancées telles que RFMOA ou DBM suivies de MS-CAGRU pour la prédiction précise en temps réel de la qualité de l'eau, ce travail repose sur des méthodes de classification traditionnelles [32].

Monirul et al. (2023) ont proposé un système embarqué basé sur Arduino pour collecter des données via des capteurs et les stocker dans une plateforme cloud sous forme de fichier CSV. La prédiction se concentre principalement sur l'évaluation de l'adéquation d'un étang pour la pisciculture, en utilisant des attributs spécifiques comme la Demande Biochimique en Oxygène (DBO), la Demande Chimique en Oxygène (DCO) et l'Oxygène Dissous (DO). Plusieurs algorithmes de machine learning ont été évalués, le modèle Random Forest obtenant la meilleure précision (94,42 %). Cependant, ce

travail est limité par son application exclusive à la pisciculture et par son nombre restreint d'attributs, contrairement à notre approche qui intègre une large gamme d'attributs environnementaux et techniques avancées pour gérer les dépendances temporelles [33].

Rodriguez et al. (2023) ont intégré des techniques d'ensemble (bagging et boosting) dans un système aquaponique piloté par l'IIoT (Industrial IoT). Le système assure une surveillance autonome et une gestion efficace des données collectées, facilitant la prise de décisions grâce au couplage de l'IIoT avec l'intelligence artificielle. À l'inverse de notre étude qui privilégie l'utilisation de l'IoT pour une collecte précise et en temps réel des données, ce travail se concentre davantage sur l'intégration avancée de l'IIoT avec l'IA et les techniques d'ensemble [34].

Karimanzira et Rauschenbach (2019) ont présenté un modèle analytique prédictif utilisant l'IoT, couplé à des systèmes ERP, SCADA et MES pour gérer des données provenant de plusieurs localisations géographiques. L'analyse économétrique, optimisée par des simulations Monte Carlo, a démontré la supériorité des approches basées sur l'IoT. Toutefois, ce travail se distingue par son intégration large de technologies diverses et par son objectif principal de démontrer la robustesse générale des prédictions plutôt qu'une spécialisation poussée dans la prédiction en temps réel de paramètres spécifiques de la qualité de l'eau, ce qui est au cœur de notre approche [35].

1.7 Conclusion

Ce premier chapitre a permis de dégager un cadre conceptuel solide en présentant l'état de l'art sur l'aquaponie intelligente et les technologies numériques qui lui sont associées. L'aquaponie, en combinant les principes de l'aquaculture et de l'hydroponie, constitue une réponse durable aux enjeux croissants liés à la sécurité alimentaire, à la gestion des ressources en eau et à la protection de l'environnement. Toutefois, l'efficacité de ce système repose sur le contrôle rigoureux de nombreux paramètres, notamment ceux liés à la qualité de l'eau.

L'intégration de l'Internet des Objets (IoT) se révèle alors essentielle, car elle permet la collecte automatisée et continue de données critiques, facilitant ainsi la supervision et la maintenance en temps réel. L'intelligence artificielle (IA), quant à elle, apporte des capacités d'analyse avancée, de détection d'anomalies et de prévision, contribuant à l'optimisation et à la prise de décision proactive dans la gestion des systèmes aquaponiques. Enfin, le cloud computing complète ce triptyque technologique en assurant la centralisation, la disponibilité et la scalabilité des services numériques.

Tous ces éléments technologiques constituent les fondations indispensables à la conception du système SISCAS. Le chapitre suivant se concentrera sur la conception technique détaillée du système, en mettant en lumière son architecture globale, ses composants physiques et logiques, ainsi que sa modélisation UML en vue de l'implémentation future.

Chapitre 2

Conception du Système SISCA

2.1 Introduction

Le projet SISCA vise une supervision intelligente pour les fermes aquaponiques, et choisit donc une architecture technique à la fois solide, flexible et capable d'absorber de futures extensions. Ce chapitre esquisse le schéma global du dispositif, en détaillant les flux entre chaque unité, les éléments matériels retenus, et la double liste des exigences fonctionnelles et non fonctionnelles qui guident le pas de conception. L'infusion de l'Internet des Objets et de routines d'allocation intelligente doit, en substance, autoriser un suivi fin, une gestion réactive du milieu et une optimisation permanente des équipements aquaponiques. À chaque étape, le lecteur est invité à visualiser comment les composants se parlent et se rectifient sans attendre.

2.2 Architecture Générale du Système

2.2.1 Description Globale du Système

L'architecture générale du système de supervision intelligente pour l'aquaponie repose sur une communication centralisée, efficace et évolutive, basée sur l'Internet des Objets (IoT) et l'intelligence artificielle. Le système est structuré autour de deux unités principales : l'unité extérieure et l'unité intérieure, reliées via le protocole MQTT à travers le réseau Wi-Fi ou Internet.

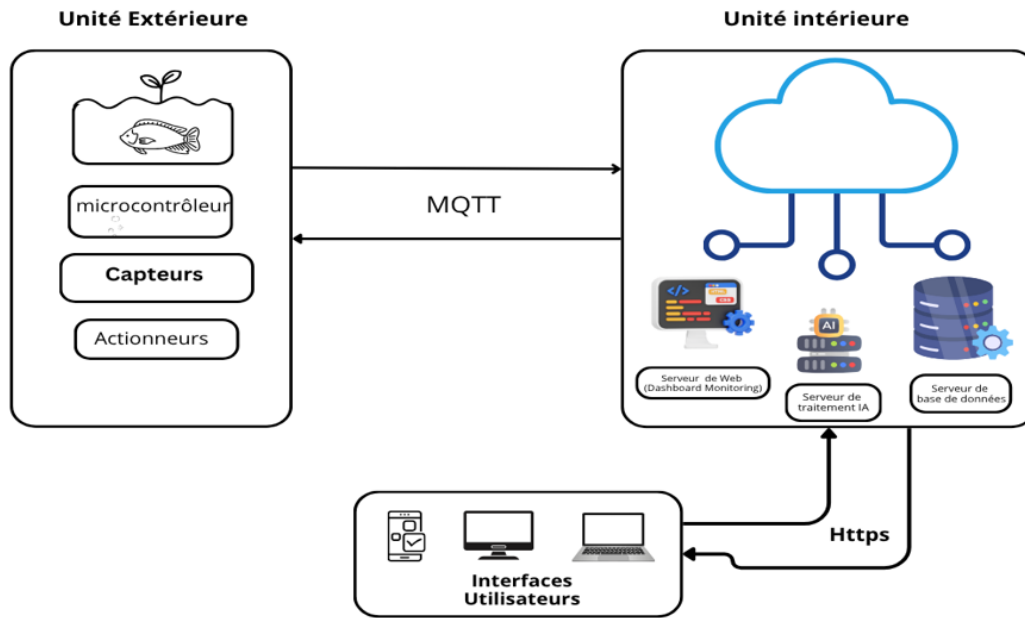


FIGURE 2.1 – L'architecture générale du système SISCA.

Unité Extérieure :

L'unité extérieure est placée au niveau des bassins et se compose :

- D'un microcontrôleur (Arduino, ESP8266) chargé de collecter les données et de piloter les actionneurs.
- D'un ensemble de capteurs surveillant les paramètres essentiels de l'environnement aquaponique : température, humidité, pH, oxygène dissous, NPK, ammoniac, nitrites, conductivité électrique (CE), turbidité.
- D'actionneurs permettant d'agir sur le selon les recommandations ou commandes reçues.

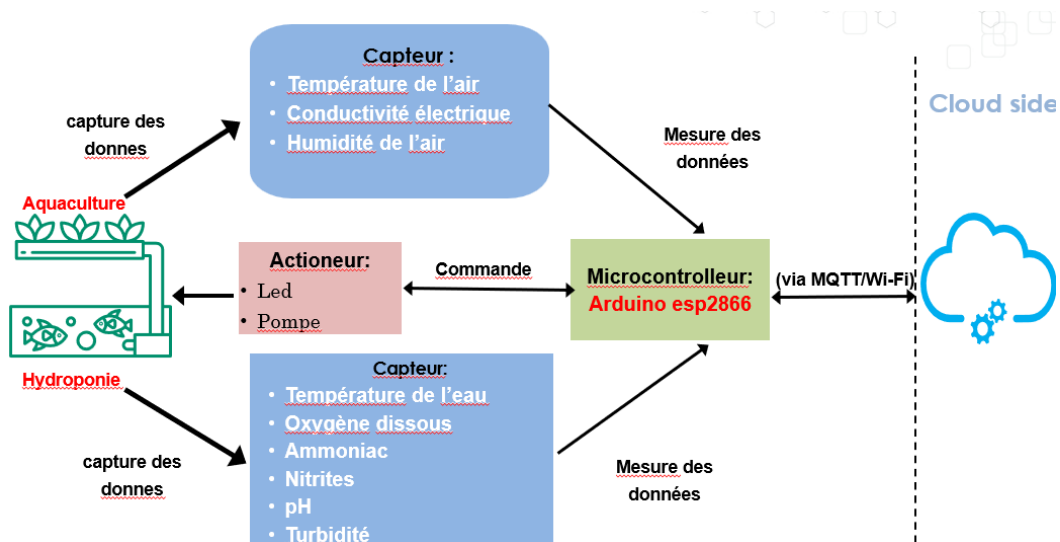


FIGURE 2.2 – L'architecture d' unité extérieure système SISCA.

Unité Intérieure :

L'unité intérieure assure le traitement avancé, la supervision et la centralisation des données. Elle intègre :

- Un serveur de base de données pour l'enregistrement et l'historisation des mesures.
- Un serveur de traitement IA chargé de l'analyse, de la prédiction.
- Un serveur Web (Dashboard Monitoring) qui permet aux utilisateurs d'accéder aux données, de superviser le système et d'interagir avec celui-ci via une interface web conviviale.
- Un système de notifications et alertes en temps réel intégré au dashboard web, informant immédiatement l'utilisateur de tout événement critique.

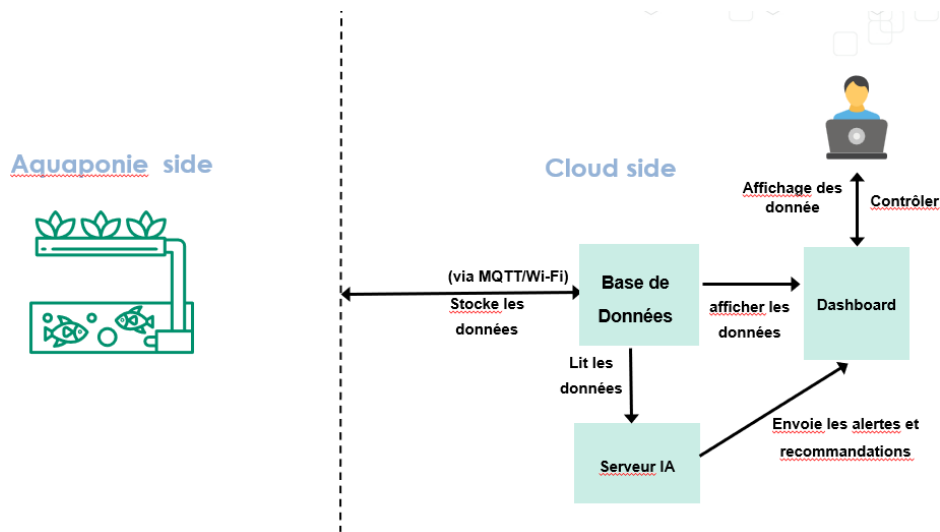


FIGURE 2.3 – L'architecture d'unité intérieure système SISCA.

Communication et Interface Utilisateur :

La communication entre l'unité extérieure et l'unité intérieure s'effectue grâce au protocole MQTT, garantissant un transfert de données rapide, fiable et sécurisé. L'utilisateur accède à toutes les fonctionnalités à partir d'un dashboard web unique, qui centralise la supervision, la visualisation des données, la gestion des alertes et le contrôle à distance des actionneurs.

2.2.2 Composants matériels principaux

2.2.2.1 Capteurs (sensors)

a) Température de l'eau – DS18B20

Le capteur DS18B20 est un capteur numérique étanche à longue portée pouvant mesurer la température entre 0 °C et 50 °C avec une précision de $\pm 0,5$ °C. Immergeable directement dans le réservoir, il sert à maintenir une température idéale pour les poissons et les bactéries nitrifiantes.



FIGURE 2.4 – Le capteur DS18B20.

b) pH de l'eau – Sonde pH analogique

Cette sonde mesure l'acidité ou la basicité de l'eau, paramètre essentiel influençant la disponibilité des nutriments pour les plantes et la santé des poissons. Une sonde typique est calibrée aux tampons pH 4,7 et 10, garantissant un cycle azoté stable.



FIGURE 2.5 – Sonde pH analogique.

c) Conductivité/TDS – Capteur analogique EC/TDS

Il mesure la conductivité électrique de l'eau, liée à la concentration en sels minéraux et nutriments. Cela permet d'évaluer si les plantes reçoivent suffisamment de nutriments ou si le milieu est appauvri.



FIGURE 2.6 – Capteur analogique EC/TDS.

d) Oxygène dissous – Sonde DO analogique

Ce capteur détecte l'oxygène dissous (DO), un indicateur clé de la qualité de l'eau pour poissons et bactéries. Un niveau de DO suffisant (> 5 ppm) est vital pour éviter stress et mortalité.



FIGURE 2.7 – Sonde DO analogique.

e) **Turbidité – Capteur turbidity analogique**

Il utilise un capteur infrarouge pour mesurer la présence de particules en suspension (déchets, algues) dans l'eau. Une turbidité élevée peut obstruer les racines et indiquer la nécessité d'un nettoyage ou changement d'eau.



FIGURE 2.8 – Capteur turbidity analogique.

f) **Ammoniac ($\text{NH}_3/\text{NH}_4^+$)**

L'ammoniac est le premier produit de la dégradation des déchets de poissons (urine, branchies, décomposition organique). C'est un toxique puissant pour les poissons si les niveaux s'accumulent. Un capteur d'ammoniac (souvent chimique ou colorimétrique) permet de détecter ces concentrations dès que le système est déséquilibré. Les bactéries nitrifiantes transforment l'ammoniac en nitrite, puis en nitrate.

FIGURE 2.9 – Ammoniac NH_3 .

g) **Nitrite ($\text{NO}_2^-/\text{NO}_3^-$)**

Le nitrite est l'intermédiaire de la nitrification, issu de la conversion de l'ammoniac par les bactéries Nitrosomonas. Il reste toxique pour les poissons et peut diminuer leur capacité à transporter l'oxygène s'il est trop élevé. Sa surveillance régulière protège les poissons et indique si le biofiltre fonctionne bien.



FIGURE 2.10 – Nitrite.

2.2.2.2 Microcontrôleur(s)

a) **Arduino Uno**

Plateforme compacte avec E/S analogiques et numériques, idéale pour la collecte des données capteurs.



FIGURE 2.11 – Arduino Uno.

b) **ESP8266**

Pour la connectivité Wi Fi (ou Bluetooth), liaison avec Arduino ou contrôleur central.



FIGURE 2.12 – ESP8266.

c) **Breadboard (plaque d'essai)**

La breadboard est une plaque d'essai sans soudure qui permet de réaliser des montages électro-

niques temporaires et modulables. Elle est essentielle lors de la phase de prototypage du système SISCA, car elle facilite l'assemblage rapide des capteurs, actionneurs et microcontrôleurs, tout en permettant de tester différentes configurations sans souder les composants. Cela rend le développement, les essais et la maintenance du système plus rapides et plus sûrs.



FIGURE 2.13 – Breadboard.

2.2.2.3 Actionneurs

a) Pompes à eau et à air

Pour circulation, aération et passage entre les réservoirs/biofiltre.



FIGURE 2.14 – Pompes à eau et à air.

b) Vannes solénoïdes / servo-valves

Pour diriger le flux d'eau (biofiltration, irrigation hydroponique).

c) Électrovannes / relais 5 V

Pilotés par les contrôleurs pour activer pompes, lumières, distributeur automatique de nourriture.

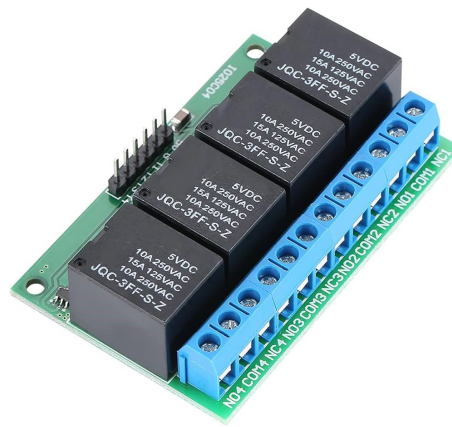


FIGURE 2.15 – Vannes solénoïdes.

d) Servo moteur

Pour nourrir les poissons selon un planning programmé.



FIGURE 2.16 – Servo moteur.

2.2.2.4 LED horticoles

La LED horticoles est un dispositif d'éclairage spécialement conçu pour fournir aux plantes la lumière nécessaire à la photosynthèse, notamment lorsque la lumière naturelle est insuffisante. Contrôlée automatiquement par le microcontrôleur selon l'heure ou le niveau de luminosité, elle permet d'optimiser la croissance des plantes dans le système aquaponique.

2.2.2.5 Caméra de surveillance

La caméra de surveillance est utilisée pour observer à distance le comportement et l'état de santé des poissons dans le bassin. Reliée au système via Wi-Fi, elle permet aux utilisateurs de visualiser en temps réel la situation sur le tableau de bord web, de détecter d'éventuels problèmes (maladies, mortalité, comportement anormal) et d'assurer une supervision visuelle continue sans intervention physique.

2.2.3 Fréquence d'Échantillonnage

TABLEAU 2.1 – Fréquence d'Échantillonnage.

Paramètre	Intervalle de collecte
Température de l'eau	Toutes les 10 minutes
pH	Toutes les 10 minutes
Oxygène dissous (O_2)	Toutes les 10 minutes
Ammoniac (NH_3/NH_4^+)	Toutes les 30 à 60 minutes
Nitrite NO_2^-	Toutes les 30 à 60 minutes
Nitrite NO_3^-	Toutes les 30 à 60 minutes
Conductivité électrique	Toutes les 10 minutes
Niveau d'eau	Toutes les 10 minutes
Turbidité	Toutes les 15 à 30 minutes
RH	Toutes les 10 minutes

2.3 Analyse des Besoins

Le développement du système SISCA nécessite une identification précise des besoins afin d'assurer une conception adaptée aux contraintes du terrain et aux objectifs du projet.

2.3.1 Besoins Fonctionnels

Les besoins fonctionnels expriment ce que le système doit faire pour remplir ses missions :

1. Surveillance en temps réel des paramètres aquaponiques.
 - Lecture automatique et périodique des capteurs (pH, température, oxygène, ammoniac, nitrates, turbidité, humidité...).
 - Transmission des données via microcontrôleur au serveur.
2. Visualisation des données à distance.
 - Affichage clair et en temps réel des valeurs sur un tableau de bord web.
 - Visualisation sous forme de graphiques (courbes, jauges, historiques...).
3. Gestion intelligente des alertes.
 - Détection automatique des dépassements de seuils critiques (pH acide, manque d'oxygène, etc.).
 - Génération de notifications/alertes visibles dans l'interface ou par email/SMS (optionnel).
4. Commande des actionneurs.
 - Activation/désactivation des pompes à eau, aérateurs, systèmes d'éclairage via l'interface web.
 - Fonctionnement automatique en fonction des données mesurées.

5. Accès multi-utilisateur (mode SaaS).
 - Authentification sécurisée par compte utilisateur.
 - Chaque utilisateur visualise uniquement ses propres données et équipements.
6. Analyse intelligente des données (IA).
 - Classification de l'état du système (Normal / Alerte / Critique).
 - Détection d'anomalies non prédéfinies par les règles.
 - Prédications sur l'évolution des paramètres critiques (optionnel).
7. Historique et exportation des données.
 - Sauvegarde horodatée de toutes les mesures.
 - Exportation en format CSV ou Excel pour analyse manuelle.

2.3.2 Besoins Non Fonctionnels

Les besoins non fonctionnels précisent les qualités attendues du système, indépendamment de ses fonctions métier.

1. Disponibilité et continuité de service.
 - Le système doit fonctionner 24h/24 et 7j/7 sans interruption.
 - Capacité à récupérer après une coupure réseau ou redémarrage (résilience).
2. Fiabilité et précision.
 - Précision des mesures capteurs et fiabilité des transmissions.
 - Limitation des fausses alertes et détection des erreurs de lecture.
3. Sécurité.
 - Sécurisation de l'accès à l'interface utilisateur (authentification, mots de passe).
 - Protection des communications entre les composants (protocole sécurisé, filtrage IP, API key).
4. Scalabilité (évolutivité).
 - Possibilité d'ajouter facilement de nouveaux capteurs, actionneurs ou utilisateurs sans reconfigurer tout le système.
5. Modularité et maintenabilité.
 - Le système doit être modulaire (séparation claire entre capteurs, IA, interface).
 - Facilité de mise à jour des composants logiciels ou matériels.
6. Compatibilité Cloud.
 - Hébergement sur un serveur distant (VPS ou plateforme cloud).
 - Accès à l'interface depuis tout navigateur connecté à Internet.

2.4 Modélisation UML du Système

2.4.1 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation est une représentation graphique permettant de visualiser le comportement fonctionnel d'un système, car il décrit l'interaction entre l'acteur et le système. Il est donc nécessaire de définir les cas d'utilisation réalisés par les utilisateurs (client et administrateur), comme illustré dans la figure suivante :

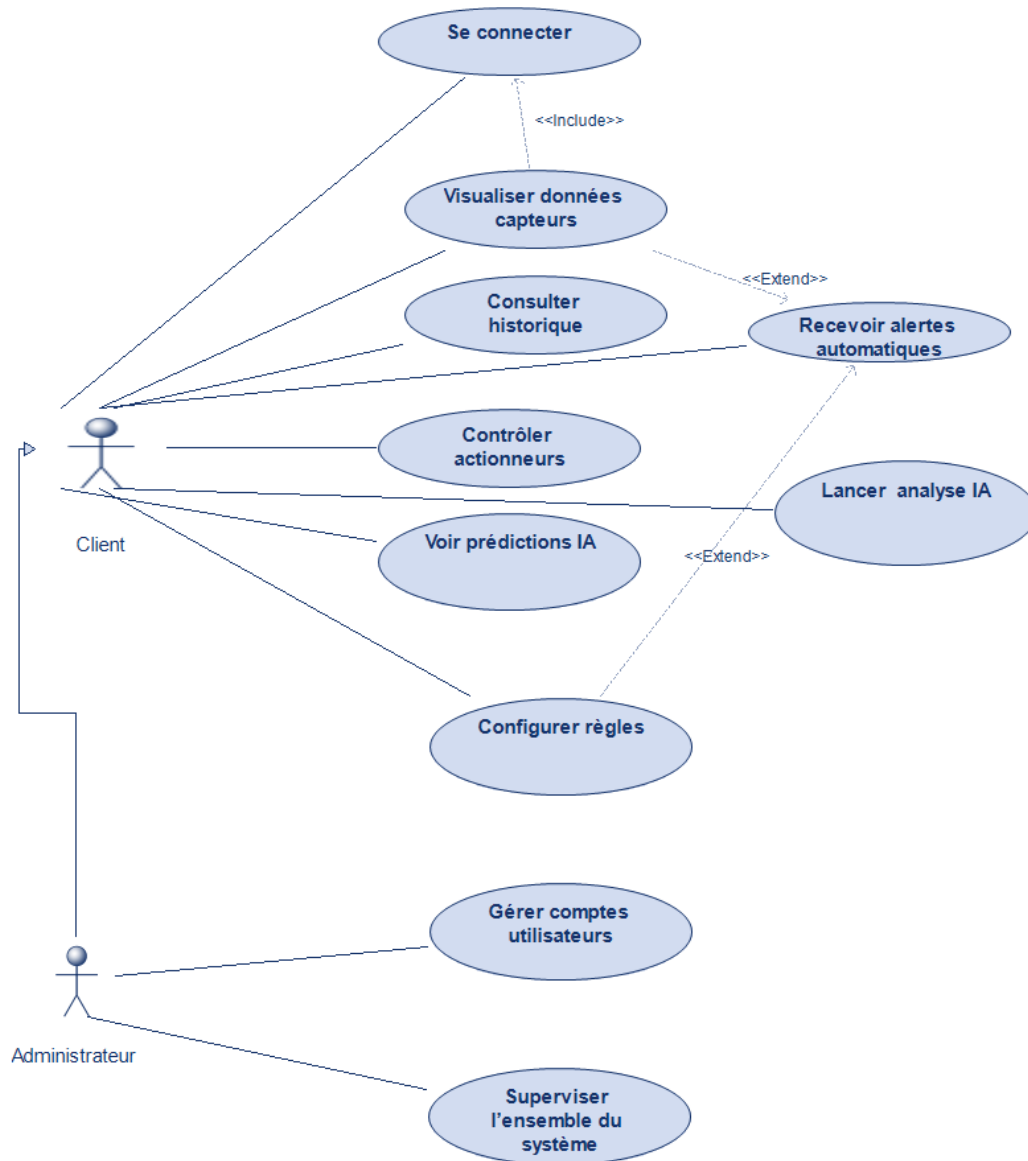


FIGURE 2.17 – Diagramme de cas d'utilisation du système SISCA.

- **Acteurs principaux :**

- **Client :**

Utilisateur final du système, généralement un responsable d'exploitation ou un technicien qui gère et supervise l'état des bassins.

— **Administrateur :**

Utilisateur avec des droits étendus permettant la gestion du système et des utilisateurs.

• **Cas d'utilisation du Client :**

1. **Se connecter :**

Le client doit s'authentifier pour accéder à l'ensemble des fonctionnalités du système.

Ce cas d'utilisation est inclus (« include ») dans toutes les autres fonctionnalités, car l'accès nécessite toujours une authentification.

2. **Visualiser données capteurs :**

Le client consulte en temps réel les valeurs remontées par les capteurs (DO, EC, pH, NH₃, température, etc.) de chaque bassin.

— Permet le suivi de l'état des paramètres essentiels.

— Peut inclure des filtres par bassin ou par période.

3. **Consulter historique :**

Le client peut afficher l'historique des mesures enregistrées par les capteurs sur différentes périodes (jour, semaine, mois).

— Facilite l'analyse et la détection de tendances ou d'anomalies.

4. **Contrôler actionneurs :**

Le client peut commander à distance certains actionneurs (pompes, valves, etc.) selon l'état du système ou ses besoins.

— L'interface propose des boutons pour activer/désactiver les équipements.

5. **Configurer règles :**

Le client définit ou modifie les seuils et règles de fonctionnement (par exemple, valeurs limites de pH, DO...).

— Ces règles sont utilisées par le système pour générer des alertes automatiques.

6. **Recevoir alertes automatiques :**

Le système notifie automatiquement le client en cas de dépassement de seuils ou d'anomalies détectées.

— Ce cas d'utilisation est **étendu** («**extend**») par la visualisation des capteurs et la configuration des règles.

7. **Lancer analyse IA :**

Le client peut lancer une analyse avancée basée sur l'intelligence artificielle pour prédire l'évolution de certains paramètres ou diagnostiquer des dysfonctionnements.

8. **Voir prédictions IA :**

Le client consulte les résultats et prédictions générés par l'IA.

— Ce cas étend («**extend**») le lancement d'analyse IA.

• **Cas d'utilisation de l'Administrateur :**

1. **Gérer comptes utilisateurs :**

L'administrateur ajoute, modifie, ou supprime les comptes utilisateurs du système.

— Gestion des rôles, des autorisations et des accès.

2. Superviser l'ensemble du système

L'administrateur a une vision globale de toutes les fonctionnalités, des historiques, et peut intervenir sur l'ensemble des paramètres du système.

• Relations et dépendances :

— «include» :

Indique qu'un cas d'utilisation nécessite systématiquement l'exécution préalable d'un autre cas (exemple : l'authentification).

— «extend» :

Indique qu'un cas d'utilisation est déclenché de façon optionnelle ou conditionnelle depuis un autre cas (exemple : la réception d'alertes dépend de la visualisation des capteurs et des règles).

Ce diagramme de cas d'utilisation offre une vue globale et détaillée des interactions possibles entre les différents profils utilisateurs et le système SISCA. Il sert de référence lors de la phase de conception et garantit que toutes les exigences fonctionnelles sont bien prises en compte dans la réalisation de la solution.

2.4.2 Diagramme de classes

La modélisation par diagramme de classes permet de représenter l'architecture objet de notre système de supervision et de contrôle aquaponique SISCA.

La figure ci-dessous illustre les principales classes du système ainsi que leurs relations.

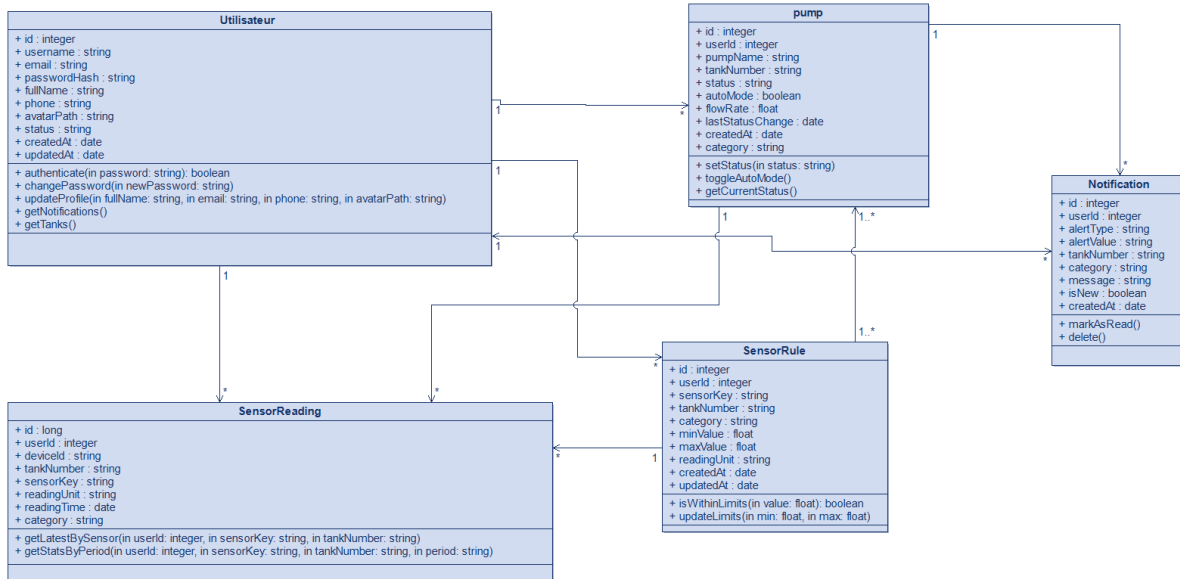


FIGURE 2.18 – Diagramme de classes du système SISCA.

• Description :

Le diagramme de classes comprend les entités principales suivantes :

- **Utilisateur** : représente un utilisateur du système (client ou administrateur). Il possède des informations personnelles (identifiant, nom, email, etc.) et des méthodes pour l'authentification, la gestion du profil, la récupération des notifications, etc. Un utilisateur est lié à plusieurs pompes, capteurs, règles et notifications.
- **Pump (Pompe)** : représente une pompe contrôlée dans le système. Chaque pompe est associée à un utilisateur et à un bassin via l'attribut tankNumber. Elle possède des caractéristiques telles que le débit, le mode automatique, le statut, etc. Une pompe peut générer plusieurs notifications en cas d'anomalie.
- **SensorReading (LectureCapteur)** : correspond à une mesure brute issue d'un capteur (pH, température, etc.), associée à un utilisateur, à un bassin et à un capteur précis (sensorKey). Ces lectures servent à la supervision en temps réel et à l'analyse historique.
- **SensorRule (RègleCapteur)** : définit les seuils (minimum/maximum) d'un capteur pour un bassin donné. Si une lecture sort de ces bornes, une notification est générée.
- **Notification** : informe l'utilisateur d'un dépassement de seuil, d'une anomalie ou d'un événement important. Elle est liée à un utilisateur, à un bassin (tankNumber), et éventuellement à une pompe (pour les alarmes liées aux actionneurs).

- **Remarque :**

L'attribut tankNumber est utilisé comme clé de liaison entre les différentes entités (pompes, règles, lectures, notifications) afin de regrouper et d'analyser les informations par bassin.

- **Associations principales :**

- Un utilisateur possède plusieurs pompes, règles, lectures et notifications.
- Une pompe peut générer plusieurs notifications.
- Chaque notification, lecture et règle est liée à un bassin via tankNumber.
- Les associations et multiplicités sont précisées sur le diagramme.

2.4.3 Diagramme de Séquence

Le diagramme de séquence permet de représenter les échanges de messages entre les acteurs externes et le système, ou entre les différentes composantes internes du système. Il offre une vue claire et structurée des étapes d'un cas d'utilisation en mettant en évidence la chronologie des interactions. Dans notre travail, nous avons utilisé des diagrammes de séquence UML afin de modéliser ces échanges et interactions entre l'utilisateur et les objets du système. Dans les sections suivantes, nous présenterons l'ensemble des diagrammes de séquence réalisés pour notre système.

Scénario 1 : "Gestion des Demandes d'Accès"

Le diagramme suivant représente le processus complet de soumission et traitement d'une demande d'accès au système SISCA.

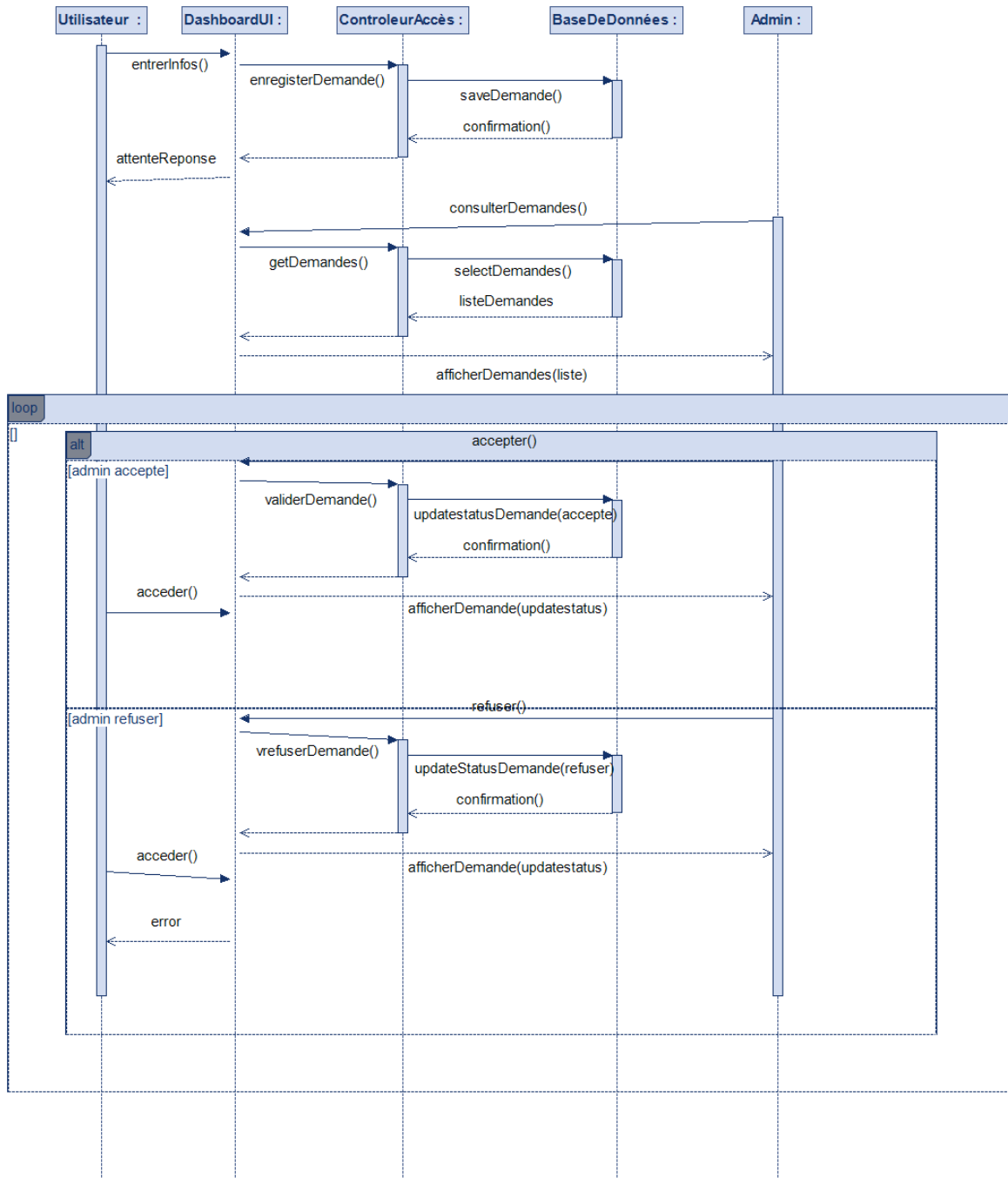


FIGURE 2.19 – Diagramme de séquence traitement des demandes d'accès.

L'utilisateur lance le processus en remplissant un formulaire via l'interface DashboardUI, qui transmet les informations à ContrôleurAccès. Ce dernier ajoute la demande à la base de données.

Le contrôleur examine ensuite périodiquement les demandes enregistrées afin de les afficher sur l'interface d'administration.

Pour permettre à l'administrateur de gérer plusieurs demandes, une boucle de blocs a été ajoutée. Deux cas sont représentés à l'intérieur de cette boîte par un bloc supplémentaire :

- **[admin accepte]** : l'utilisateur peut désormais accéder à son espace après que la demande a été

vérifiée et que son statut est passé à « accepté ».

- **[admin refuse]** : l'utilisateur reçoit un message d'erreur, la demande est refusée et le statut passe à « refusé ».

Ce modèle clarifie la répartition des responsabilités au sein du système ainsi que la logique asynchrone.

Scénario 2 : "Surveillance et interaction utilisateur"

Ce scénario représente une interaction utilisateur typique dans un système de supervision aquaponique intelligent. Il démontre la centralisation des fonctions essentielles de surveillance et de contrôle dans une interface unique, tout en assurant l'intégrité des échanges via une architecture en couches.

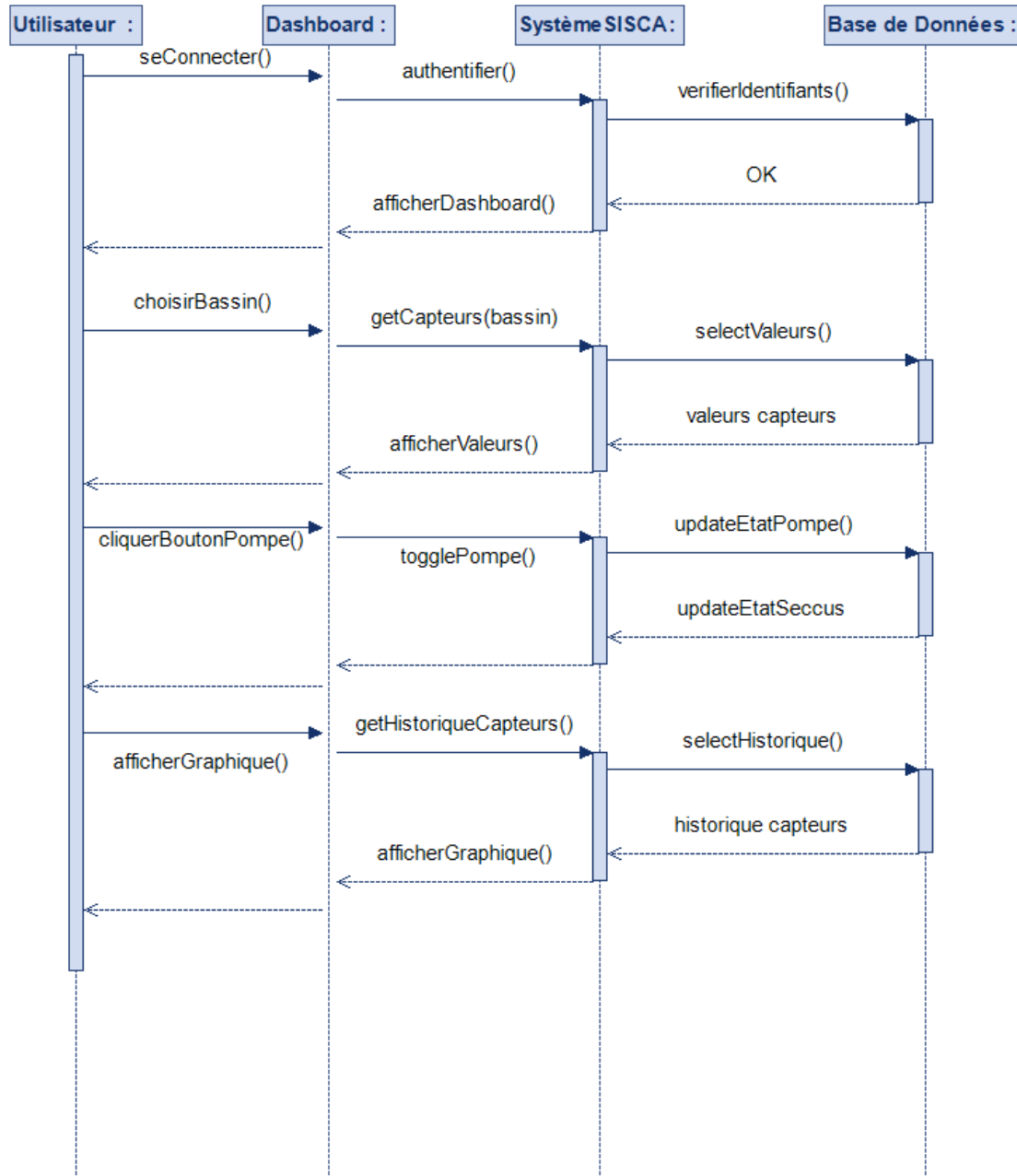


FIGURE 2.20 – Diagramme de séquence surveillance et interaction utilisateur.

Il regroupe les principales interactions disponibles à travers le tableau de bord, notamment :

1. **Connexion de l'utilisateur** (authentification),
2. **Choix du bassin à superviser**,
3. **Récupération et affichage des valeurs des capteurs**,
4. **Contrôle des pompes (activation/désactivation)**,
5. **Affichage de l'historique sous forme graphique**.

Ce scénario global met en évidence les échanges de messages entre les quatre principaux acteurs :

1. **L'utilisateur** (acteur externe),
2. **L'interface Dashboard** (couche de présentation),
3. **Le système SISCA** (couche logique de traitement),
4. **La base de données** (couche de persistance).

2.5 Conclusion

Ce deuxième chapitre a permis de poser les bases techniques du projet SISCA en détaillant son architecture globale, ses composants matériels essentiels, ainsi que les besoins fonctionnels et non fonctionnels auxquels il doit répondre. La conception repose sur une structure modulaire, intégrant des capteurs intelligents, des microcontrôleurs connectés, et une plateforme web interactive conçue pour assurer une supervision fine et réactive du système aquaponique.

Les choix d'architecture réalisés garantissent la flexibilité, l'évolutivité et la résilience du système face aux contraintes du terrain. En complément, une modélisation UML rigoureuse a été élaborée pour formaliser les cas d'usage, les relations entre les entités et les scénarios d'interaction utilisateur-système, préparant ainsi efficacement la phase de développement.

Dans le chapitre suivant, nous passerons à la phase d'implémentation, en présentant les outils de développement utilisés, l'intégration des modèles d'intelligence artificielle, les résultats expérimentaux, ainsi que les interfaces de la plateforme SISCA.

Chapitre 3

Implementation

3.1 Introduction

Ce dernier chapitre présente la mise en œuvre du système intelligent de gestion aquaponique SISCA (Aquasmart). Nous commençons par décrire l’environnement technique, en détaillant les principaux langages de programmation, les outils de développement et l’architecture globale du système. Le processus d’implémentation est présenté de manière progressive, en commençant par la configuration du serveur et de la base de données, puis en abordant le développement du firmware embarqué pour l’acquisition des données des capteurs et le contrôle des actionneurs. Le chapitre traite ensuite de la conception et de l’intégration des services backend en PHP, ainsi que de la réalisation du tableau de bord web interactif avec des technologies frontend modernes. Nous évoquons également la communication en temps réel entre les composants matériels et logiciels, l’utilisation d’outils de visualisation des données, ainsi que l’implémentation des principales fonctionnalités du système telles que la gestion des notifications et des utilisateurs. Enfin, une vue d’ensemble de la structure de la plateforme et de l’interface utilisateur est proposée, illustrant comment le système permet une supervision fluide et une automatisation intelligente des environnements aquaponiques.

3.2 Langages de programmation

1. HTML :

Le **HyperText Markup Language (HTML)** est la pierre angulaire du Web. Il définit la signification et la structure du contenu web. Les autres technologies, comme CSS pour l’apparence et JavaScript pour le comportement, viennent compléter le rendu des pages HTML [36]. Dans le projet Aquasmart, HTML sert à définir la mise en page, intégrer des éléments multimédias et structurer les interfaces utilisateur pour les tableaux de bord, les paramètres, les notifications et les analyses. HTML fournit la structure sémantique de toutes les pages et composants du frontend.

2. CSS :

Les **Cascading Style Sheets (CSS)** constituent un langage de feuilles de style utilisé pour

décrire la mise en forme d'un document écrit en HTML ou XML (par exemple SVG, MathML, XHTML). CSS détermine la manière dont les éléments sont affichés à l'écran, en impression, à l'oral ou sur d'autres supports. C'est l'un des langages clés du Web ouvert, standardisé par le W3C [36].

3. JavaScript :

JavaScript (JS) est un langage de programmation léger, interprété ou compilé à la volée (Just In Time), avec des fonctions de première classe. Bien qu'il soit surtout connu comme langage de script côté client pour les pages Web, il est également utilisé dans des environnements hors navigateur, tels que Node.js, Apache CouchDB ou Adobe Acrobat. JavaScript est un langage prototype based, multi paradigme, mono thread, dynamique, qui supporte les styles de programmation orientée objet, impérative et déclarative (comme la programmation fonctionnelle) [36].

4. Tailwind CSS :

Tailwind CSS est un framework utilitaire qui s'est imposé comme une référence du développement web moderne. Contrairement à d'autres frameworks comme Bootstrap, Tailwind mise exclusivement sur des classes utilitaires « atomiques », ce qui permet aux développeurs de créer des designs sur mesure directement dans le code HTML. Cette approche a profondément changé la façon de concevoir les interfaces, en offrant à la fois flexibilité et rapidité de développement [37].

5. Chart.js :

est une bibliothèque JavaScript open source permettant de créer des graphiques interactifs et adaptatifs. Dans Aquasmart, Chart.js est utilisé pour visualiser les données des capteurs sous forme de séries temporelles, permettant aux utilisateurs d'analyser la température, le pH, l'ammoniac et d'autres paramètres essentiels via des graphiques interactifs sur le tableau de bord.

6. SQL :

Le **Structured Query Language (SQL)** est l'un des plus anciens langages utilisés pour interagir avec des bases de données relationnelles, et le plus répandu. Il permet d'interroger, mettre à jour et organiser les données stockées dans une base, d'ajouter ou de supprimer des éléments, et même de modifier sa structure et ses droits d'accès. SQL peut traiter des volumes de données bien plus importants que ceux d'un tableur classique (comme Microsoft Excel), restant performant même sur des millions ou des milliards d'enregistrements [38].

7. PHP :

PHP (acronyme de « **PHP** : Hypertext Preprocessor », à l'origine « Personal Home Page ») est un langage de script populaire utilisé pour générer dynamiquement des pages Web. Il combine une syntaxe tirée du C, de Java et de Perl, et s'intègre directement dans le code HTML pour être exécuté côté serveur. PHP est fréquemment utilisé pour extraire des données d'une base MySQL et les afficher sur la page web. D'abord conçu pour les pages personnelles, il est désormais omniprésent via son exécution sur serveur et largement supporté [39].

8. Python :

Python est un langage de programmation interprété, orienté objet et de haut niveau, avec une typage dynamique. Ses structures de données intégrées et sa syntaxe simple favorisent le dévelop-

pement rapide d'applications et facilitent la maintenance grâce à une meilleure lisibilité. Python supporte les modules et packages, ce qui encourage la modularité et la réutilisation du code. L'interpréteur et la bibliothèque standard sont disponibles gratuitement sur toutes les plateformes majeures [40].

9. **C++ :**

Le C++ est le principal langage de programmation utilisé pour développer le firmware des microcontrôleurs Arduino du projet Aquasmart. Ce firmware gère l'acquisition de données en temps réel provenant de capteurs et contrôle des actionneurs tels que des pompes et des aérateurs. Le code C++ permet un accès matériel efficace et prend en charge les protocoles de communication réseau (tels que MQTT ou HTTP) via des modules comme l'ESP8266. L'utilisation de ces langages garantit des performances et une fiabilité élevées pour les tâches IoT embarquées.

10. **API :**

Une API (Application Programming Interface) est une interface qui permet à deux programmes ou applications de dialoguer entre eux. Elle définit un ensemble de règles et de formats d'échange pour que des logiciels différents puissent s'envoyer des données, accéder à des fonctionnalités ou intégrer des services, le tout de façon automatisée et sécurisée, sans intervention humaine directe [41].

3.3 Outils de développement

1. **Arduino IDE :**

L'Arduino Integrated Development Environment (IDE), ou logiciel Arduino, propose un éditeur de texte pour l'écriture du code, une zone de messages, une console de texte, une barre d'outils regroupant les fonctions courantes, ainsi qu'un ensemble de menus. Cet environnement permet de se connecter au matériel Arduino afin de téléverser les programmes et de communiquer avec les cartes électroniques [42].

2. **Visual Studio Code :**

Visual Studio Code est un éditeur de code source léger mais puissant, disponible pour Windows, macOS et Linux. Il offre une prise en charge intégrée de JavaScript, TypeScript et Node.js, et dispose d'un vaste écosystème d'extensions pour de nombreux autres langages et environnements d'exécution, tels que C++, C#, Java, Python, PHP, Go, et .NET [43].

3.4 Implémentation du modèle prédictif avec le Deep Learning

L'intelligence artificielle intégrée dans mon système d'aquaponie intelligente a pour objectif de prédire les paramètres de qualité de l'eau à venir à partir des données historiques des capteurs. En anticipant des valeurs telles que le pH, l'ammoniaque, l'oxygène dissous et d'autres, l'IA permet une détection précoce des déséquilibres ou anomalies qui pourraient nuire à la santé des poissons et des plantes.

Grâce à ces prévisions, le système peut activer automatiquement les pompes ou générer des notifica-

tions pour alerter l'utilisateur, facilitant ainsi une gestion proactive et optimisée de l'environnement aquaponique. Cela contribue à maintenir un équilibre écologique optimal, à réduire les interventions manuelles et à améliorer la durabilité et l'efficacité globale du système.

3.4.1 Définition du Deep Learning

Le deep learning, ou apprentissage profond, constitue une branche avancée du machine learning. Cette technologie repose sur l'utilisation de réseaux de neurones artificiels capables d'imiter le fonctionnement du cerveau humain. Ces réseaux se composent de plusieurs dizaines, voire centaines de couches de neurones, chaque couche traitant et transmettant l'information à la suivante pour permettre des analyses et des prises de décision complexes [44].

3.4.2 Modèle Utilisé

3.4.2.1 Modèle principal : GRU (Gated Recurrent Unit)

Type : Réseau de neurones récurrent (RNN), plus précisément une variante appelée GRU. La GRU est une architecture de réseau de neurones récurrents optimisée pour le traitement des données séquentielles. Elle utilise des mécanismes de portes pour mémoriser les informations importantes sur de longues séquences, tout en restant plus simple et plus rapide à entraîner que les LSTM. Les GRU sont largement utilisées pour la prédiction et l'analyse de séries temporelles [45].

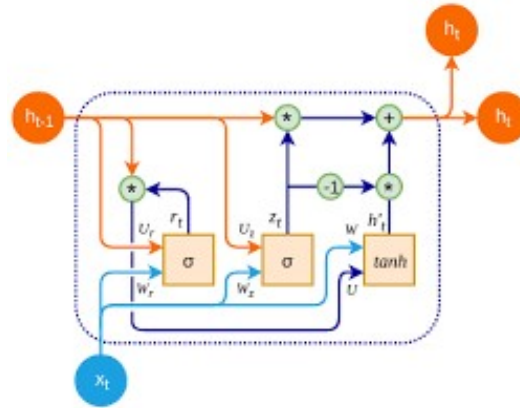


FIGURE 3.1 – Architecture GRU.

3.4.2.2 Modèle secondaire : Random Forest Regressor (RF)

Random Forest est un algorithme d'apprentissage supervisé très populaire, utilisé pour la classification et la régression. Random Forest est largement utilisé par les data scientists pour extraire des informations pertinentes et guider la prise de décision à partir de données volumineuses [46].

Justification :

- Les données capteurs sont des séries temporelles : chaque mesure dépend des valeurs passées (dépendances temporelles).

- GRU est efficace pour modéliser ces dépendances, mieux que les réseaux classiques car il gère le problème du gradient qui disparaît grâce à ses portes (gating).
- Moins complexe que LSTM, GRU est plus rapide à entraîner tout en conservant une bonne capacité à apprendre les séquences.
- Le RF capture des relations non-linéaires supplémentaires dans les prédictions du GRU.
- C'est une méthode d'ensemble robuste qui réduit l'erreur et améliore la stabilité des prédictions.

3.4.3 Méthodologie

1. Préparation et nettoyage des données :

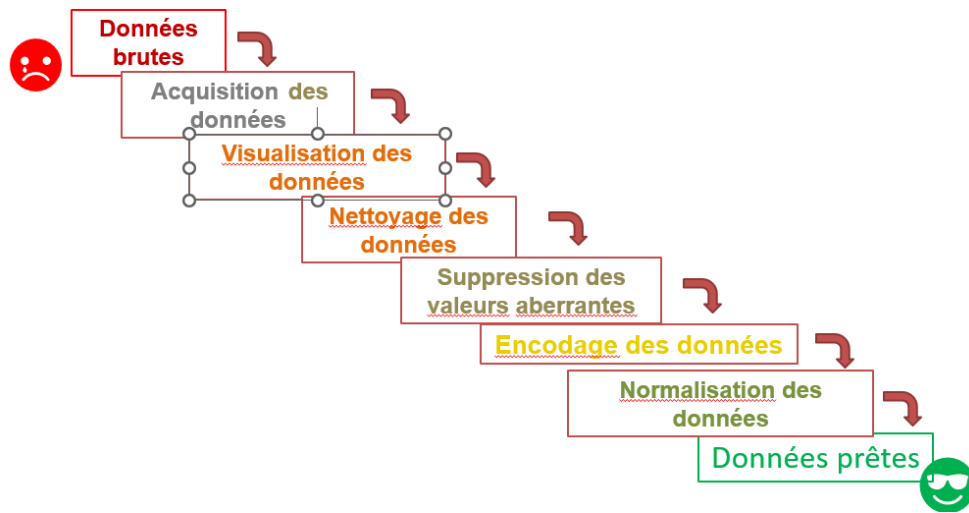


FIGURE 3.2 – Préparation et nettoyage des données.

- Collecte systématique des données issues des capteurs IoT.
 - Validation rigoureuse et élimination des anomalies, données aberrantes et valeurs manquantes.
2. Normalisation des données :
 - Mise à l'échelle des données pour optimiser la performance d'apprentissage et accélérer la convergence du modèle.
 3. Génération des séquences temporelles :
 - Structuration minutieuse des données en séries temporelles, adaptées spécifiquement à l'apprentissage par GRU.
 - Séquences temporelles définies en fonction des intervalles optimaux pour capter efficacement les tendances.
 4. Entraînement du modèle GRU :
 - Apprentissage détaillé des dépendances temporelles à partir des données historiques.
 - Optimisation approfondie du modèle via des techniques avancées comme l'arrêt précoce (Early Stopping) pour prévenir le surapprentissage.

5. Fusion des prédictions via une Forêt Aléatoire :

- Utilisation stratégique des sorties du GRU comme entrées pour la Forêt Aléatoire.
- Agrégation des résultats pour renforcer les prédictions finales et gérer les non-linéarités complexes.

6. Évaluation de la Performance et Prédiction pour la Prochaine Étape.

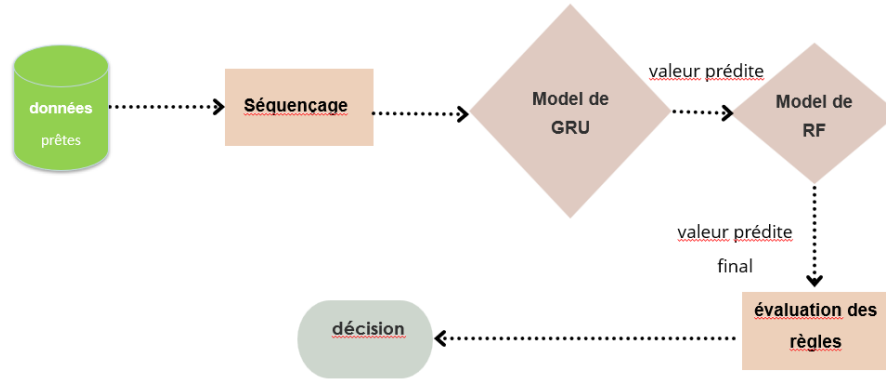


FIGURE 3.3 – Méthodologie de modèle.

3.4.4 Résultats expérimentaux : comparaisons des valeurs réelles et prédites

Interprétation des graphiques :

3.4.4.1 Comparaison Réelles vs Prédites — pH

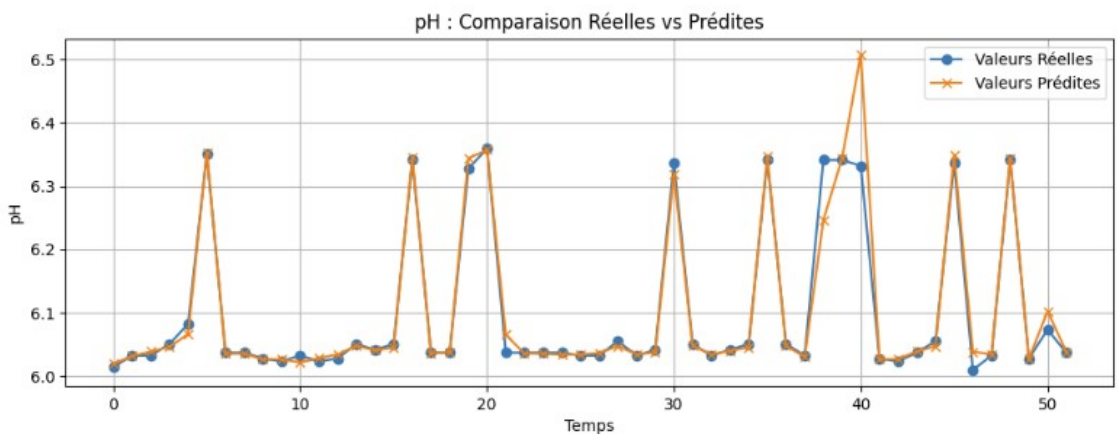


FIGURE 3.4 – Comparaison Réelles-Predites.

- Le graphique montre une forte superposition entre les valeurs réelles et prédites, traduisant une excellente capacité du modèle à apprendre la dynamique du pH.
- Les légères divergences visibles restent minimales.

3.4.4.2 Comparaison Réelles vs Prédites — Température — Turbidité

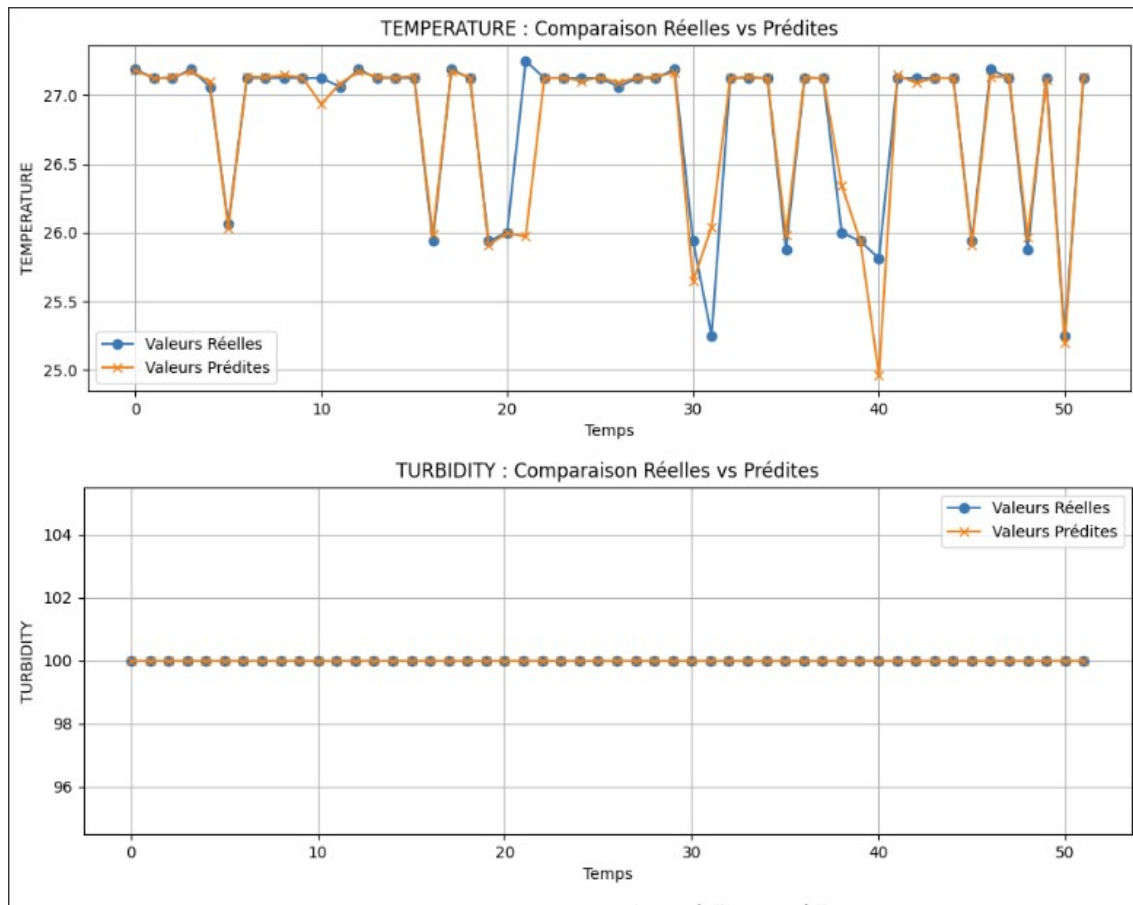


FIGURE 3.5 – Comparaison Réelles-Predites.

- On observe une très bonne concordance entre les courbes réelle et prédite, les points s'alignent presque parfaitement.
- Cela confirme le bon score de R^2 et la faible erreur.
- Les deux courbes se confondent totalement, expliquant le $R^2 = 1.0$ et les erreurs nulles.
- Cela peut s'expliquer par un manque de variation dans la série temporelle de turbidité, rendant la prédiction « facile » pour le modèle.

3.4.5 Analyse des performances du modèle de prédiction

La table de résultats présente trois indicateurs classiques d'évaluation des modèles de régression :

- MAE (Mean Absolute Error).
- RMSE (Root Mean Square Error)
- R^2 (coefficient de détermination).

index	Feature	MAE	RMSE	R ²
0	TEMPERATURE	0.088	0.25	0.813
1	TURBIDITY	0.0	0.0	1.0
2	DISOLVED OXYGEN	1.195	1.706	0.744
3	pH	0.011	0.029	0.945
4	AMMONIA	365.319	1413.31	-0.464
5	NITRATE	60.167	177.718	0.746

FIGURE 3.6 – Trois indicateurs performances du modèle.

Pour chaque paramètre, on observe les résultats suivants :

- TEMPÉRATURE :

- MAE = 0.088, RMSE = 0.25, R² = 0.813

Ces valeurs indiquent que le modèle prédit la température avec une grande précision (faible erreur, R² élevé).

- TURBIDITÉ :

- MAE et RMSE = 0.0, R² = 1.0

Les prédictions sont parfaitement alignées avec les valeurs réelles, ce qui traduit soit un modèle très performant, soit un manque de variabilité dans les données (à vérifier sur la courbe).

- OXYGÈNE DISSOUT :

- MAE = 1.195, RMSE = 1.706, R² = 0.744

Le modèle a une précision correcte, mais moins élevée que pour la température ou le pH.

- pH :

- MAE = 0.011, RMSE = 0.029, R² = 0.945

Prédiction extrêmement précise, le modèle parvient à suivre parfaitement l'évolution du pH.

- AMMONIAQUE :

- MAE = 365.319, RMSE = 1413.31, R² = -0.464

Erreurs très élevées et R² négatif, ce qui signifie que le modèle n'est pas capable de prédire correctement ce paramètre (pire qu'un modèle naïf).

- NITRATE :

- MAE = 60.167, RMSE = 177.718, R² = 0.746

Résultats satisfaisants, mais avec une erreur absolue relativement importante.

3.4.6 hyperparamètres du modèle IA

Pour concevoir un modèle de prédiction performant des paramètres de qualité de l'eau dans notre système aquaponique intelligent, nous avons opté pour un modèle hybride combinant un réseau récurrent à mémoire courte de type GRU (Gated Recurrent Unit) et un modèle de forêt aléatoire (Random Forest).

```

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GRU, Dense, Dropout
from tensorflow.keras.callbacks import EarlyStopping

model_gru = Sequential([
    GRU(64, return_sequences=True, input_shape=(X_train.shape[1], X_train.shape[2])),
    Dropout(0.2),
    GRU(32),
    Dropout(0.2),
    Dense(len(features))
])

model_gru.compile(optimizer='adam', loss='mse', metrics=['mae'])
early_stop = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
model_gru.fit(X_train, y_train, epochs=30, batch_size=32, validation_split=0.2, callbacks=[early_stop], verbose=1)
from sklearn.ensemble import RandomForestRegressor

y_train_gru_pred = model_gru.predict(X_train)
y_test_gru_pred = model_gru.predict(X_test)

rf = RandomForestRegressor(n_estimators=50)
rf.fit(y_train_gru_pred, y_train)
y_pred_fusion = rf.predict(y_test_gru_pred)

```

FIGURE 3.7 – Fusion GRU + Random Forest.

Le modèle GRU utilise plusieurs hyperparamètres clés qui ont été choisis et ajustés en fonction des caractéristiques spécifiques de notre jeu de données temporelles :

- Taille de la fenêtre temporelle (window = 24) : Cette taille correspond à une séquence de 24 pas de temps. Ce choix permet au modèle de capter des dépendances à court terme dans les données, en se basant sur les variations observées sur une journée complète, ce qui est crucial pour des paramètres environnementaux qui évoluent souvent avec des cycles journaliers.
- Unités dans les couches GRU (64 unités dans la première couche, 32 dans la deuxième) : Ces valeurs déterminent la capacité d'apprentissage du modèle. Une première couche plus large (64 neurones) permet d'extraire des caractéristiques complexes, tandis que la seconde couche (32 neurones) affine ces informations avant la sortie. Ce choix équilibre la complexité du modèle et le risque de surapprentissage.
- Taux de Dropout (0.2) : Afin d'éviter le surapprentissage, nous appliquons un dropout de 20 % sur chaque couche GRU. Cela consiste à désactiver aléatoirement une fraction des neurones pendant l'entraînement, ce qui améliore la généralisation du modèle.
- Optimiseur Adam : Cet optimiseur adaptatif est choisi pour sa rapidité de convergence et sa robustesse, facilitant l'ajustement des poids du réseau lors de l'apprentissage.
- Batch size (32) : Un lot de 32 échantillons est utilisé à chaque mise à jour des poids, offrant un bon compromis entre stabilité de la descente de gradient et temps de calcul.
- Nombre d'époques (30) avec EarlyStopping (patience=5) : L'entraînement est limité à 30 itérations sur l'ensemble des données, mais s'arrête plus tôt si la performance sur l'ensemble de validation ne s'améliore pas pendant 5 époques consécutives. Cette méthode évite le surapprentissage et réduit le temps d'entraînement.
- Validation split (20 %) : 20 % des données d'entraînement sont utilisées pour évaluer la performance du modèle à chaque époque, permettant de détecter le surapprentissage.

3.4.7 Fonctionnement du Système SISCA

1. **Collecte des données** : Acquisition des données en temps réel depuis les capteurs (température, pH, oxygène dissous, conductivité électrique, ammoniac, etc.) placés dans les bassins.
2. **Préparation des données** : Nettoyage, normalisation et structuration des données collectées pour faciliter leur exploitation par le modèle d'IA.
3. **Entraînement du modèle** : Utilisation d'un algorithme de Deep Learning (GRU) pour apprendre les patterns et relations entre les différents paramètres sur des données historiques.
4. **Prédiction des valeurs futures** : Le modèle entraîné prédit les valeurs des paramètres clés à venir, permettant d'anticiper l'évolution de la qualité de l'eau.
5. **Évaluation des règles métier** : Comparaison des prédictions avec les seuils prédéfinis pour chaque paramètre (valeurs normales ou critiques). Détection des anomalies ou des situations à risque.
6. **Génération de notifications et actions** : En cas d'anomalies détectées, le système envoie des alertes aux utilisateurs et peut déclencher automatiquement des actions correctives, comme l'activation de pompes ou d'autres actionneurs.
7. **Supervision et amélioration continue** : Surveillance continue des performances du modèle et mise à jour régulière avec de nouvelles données pour améliorer la précision des prédictions et la fiabilité du système.

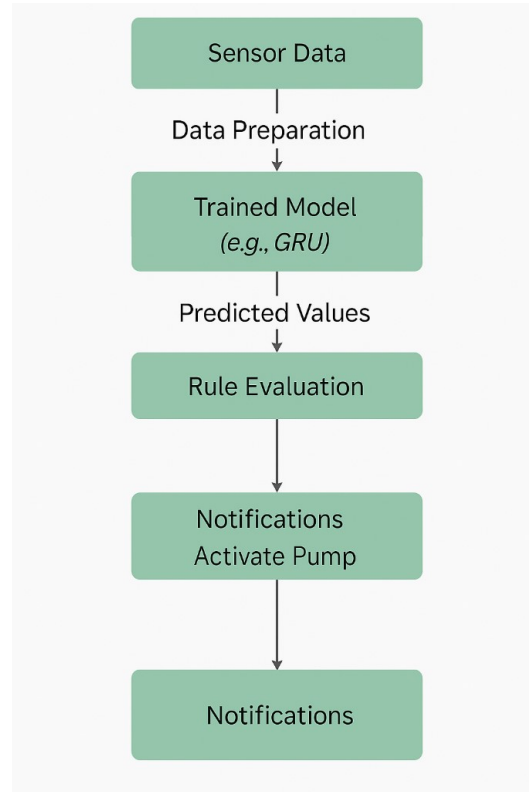


FIGURE 3.8 – Fonctionnement d'IA in SISCA.

3.4.7.1 Quelques fonctions du système

Développé assure la surveillance environnementale via des capteurs (eau, température, humidité) et permet le contrôle à distance du moteur (pompe ou autre) à partir d'un tableau de bord Web. Les capteurs envoient leurs mesures à des intervalles différents, tandis que l'ESP8266 interroge régulièrement le serveur pour exécuter toute commande utilisateur.

Configuration des composants, des pins et des intervalles Points :



```

Sketch_Water_Sensor | Arduino 1.8.19
File Edit Sketch Tools Help

Sketch_Water_Sensor
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266HTTPClient.h>
#include <DHT.h>

// Identifiants WiFi
const char* ssid = "iPhone de oussama";
const char* password = "1123456789";

// URL des scripts PHP
const char* serverDataUrl = "http://172.20.10.3/water_project.php"; // Envoi des capteurs
const char* serverCommandUrl = "http://172.20.10.3/get_motor_command.php"; // Lecture commande moteur

// Capteurs
#define WATER_SENSOR_PIN A0
#define DHTPIN D1
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

// Moteur
#define MOTOR_PIN D2

// Timers
unsigned long lastWaterSend = 0;
unsigned long lastDHTSend = 0;
unsigned long lastCommandCheck = 0;

const unsigned long intervalWater = 600000; // 10 minutes
const unsigned long intervalDHT = 300000; // 5 minutes
const unsigned long intervalCommand = 10000; // 10 secondes

```

FIGURE 3.9 – Fonction Configuration des composants, des pins et des intervalles Points.

- Inclusion des bibliothèques nécessaires : WiFi, HTTPClient, DHT.
- Déclaration des identifiants WiFi pour la connexion.
- Définition des URLs PHP pour l'envoi des données capteurs (`water_project.php`) et pour la lecture des commandes moteur (`get_motor_command.php`).
- Définition des broches :
 - Capteur d'eau connecté à A0.
 - Capteur DHT11 (température/humidité) connecté à D1.
 - Moteur relié à D2 (via relais).
- Déclaration des minuteries avec `millis()` pour exécuter différentes tâches à intervalles définis :
 - `intervalWater = 600000` → 10 min.
 - `intervalDHT = 300000` → 5 min.
 - `intervalCommand = 10000` → 10 sec

Envoi des données du capteur d'eau :

- La fonction `analogRead(WATER_SENSOR_PIN)` lit la valeur du capteur d'eau.
- `moisture_5v` est simulé comme 1023 (pleine échelle 5V).
- Une requête POST est envoyée toutes les 10 minutes au script PHP pour stocker ces données.
- Utilisation de HTTPClient pour envoyer les données formatées avec `x-www-form-urlencoded`.

— Affichage de la réponse du serveur via le moniteur série.

```
void loop() {
  unsigned long currentMillis = millis();
  WiFiClient client;

  // Envoi données capteur eau (10 min)
  if (currentMillis - lastWaterSend >= intervalWater) {
    int waterValue = analogRead(WATER_SENSOR_PIN);
    int moisture_5v = 1023;

    if (WiFi.status() == WL_CONNECTED) {
      HTTPClient http;
      http.begin(client, serverDataUrl);
      http.addHeader("Content-Type", "application/x-www-form-urlencoded");

      String postData = "water_value=" + String(waterValue) +
        "&alert=" + alertMessage +
        "&moisture_5v=" + String(moisture_5v);

      int httpResponseCode = http.POST(postData);
      if (httpResponseCode > 0) {
        Serial.println("Eau envoyée : " + http.getString());
      }
      http.end();
    }

    lastWaterSend = currentMillis;
  }
}
```

FIGURE 3.10 – Fonction Envoi des données du capteur d'eau.

Lecture des commandes moteur depuis le serveur :



```
sketch_water_sensor | Arduino 1.8.19
File Edit Sketch Tools Help

sketch_water_sensor

// Lecture commande moteur (toutes les 10 sec)
if (currentMillis - lastCommandCheck >= intervalCommand) {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    http.begin(client, serverCommandUrl);

    int httpCode = http.GET();
    if (httpCode == 200) {
      String motorState = http.getString();
      motorState.trim();

      if (motorState == "on") {
        digitalWrite(MOTOR_PIN, HIGH);
        Serial.println("Moteur ACTIVÉ");
      } else if (motorState == "off") {
        digitalWrite(MOTOR_PIN, LOW);
        Serial.println("Moteur DÉACTIVÉ");
      } else {
        Serial.println("Commande moteur inconnue : " + motorState);
      }
    } else {
      Serial.println("Erreur lecture commande moteur");
    }

    http.end();
  }

  lastCommandCheck = currentMillis;
}
}
```

FIGURE 3.11 – Fonction Lecture des commandes moteur depuis le serveur.

— Une requête GET est envoyée au fichier `get_motor_command.php`.

— La réponse "on" ou "off" détermine l'état du moteur :

- `digitalWrite(MOTOR_PIN, HIGH)` → moteur activé.
- `digitalWrite(MOTOR_PIN, LOW)` → moteur désactivé.

— Messages de confirmation affichés dans le moniteur série.

3.5 Structure de la base de données

La plateforme Aquasmart repose sur une base de données relationnelle robuste, conçue pour centraliser et organiser l'ensemble des informations nécessaires au bon fonctionnement du système. Cette base de données, nommée `aquasmart_db`, regroupe plusieurs tables principales assurant la gestion des utilisateurs, la collecte des mesures, le contrôle des actionneurs et la gestion des alertes.

3.5.1 Table des utilisateurs (`users`)

La table `users` recense l'ensemble des comptes créés sur la plateforme. Chaque enregistrement contient un identifiant unique, un nom d'utilisateur, une adresse email, un mot de passe sécurisé (haché), un numéro de téléphone, un statut (actif/inactif), un rôle (par exemple : utilisateur, administrateur) ainsi qu'un chemin vers un éventuel avatar de profil.

Grâce à cette table, il est possible de gérer l'inscription, l'authentification, l'activation ou la désactivation d'utilisateurs, ainsi que l'attribution de rôles pour la gestion des droits d'accès.

3.5.2 Table des règles capteurs (`sensor_rules`)

La table `sensor_rules` permet de définir, pour chaque bassin et chaque utilisateur, les seuils minimum et maximum associés à chaque capteur (pH, température de l'eau, conductivité, ammoniac, etc.). Chaque règle précise le type de capteur, les bornes acceptables, l'unité de mesure et la date de création. Cette configuration dynamique permet au système de générer des alertes personnalisées dès qu'une mesure sort de l'intervalle autorisé.

3.5.3 Table des relevés de capteurs (`sensor_readings`)

Toutes les données environnementales mesurées sont stockées dans la table `sensor_readings`. Chaque entrée comprend l'identifiant utilisateur, le dispositif concerné, le numéro du bassin, le type de capteur, la valeur mesurée, l'unité, ainsi que la date et l'heure du relevé.

Cette structuration permet d'assurer la traçabilité complète de l'évolution des paramètres dans chaque bassin, et d'alimenter les modules statistiques et analytiques de la plateforme.

3.5.4 Table des pompes/actionneurs (`pumps`)

La table `pumps` référence l'ensemble des pompes associées à chaque utilisateur et bassin. Elle contient des informations telles que le nom de la pompe, son état (marche/arrêt), le mode de fonctionnement (manuel/automatique), le débit, et l'historique des changements d'état.

Cela permet à la plateforme de proposer un contrôle en temps réel des actionneurs, avec historique des interventions.

3.5.5 Table des notifications (`notifications`)

La gestion des alertes et messages système est assurée par la table `notifications`. Chaque notification comporte un identifiant, l'utilisateur concerné, le type d'alerte, la valeur seuil, un message descriptif,

le statut de lecture, et la date de création. Cette organisation facilite la remontée en temps réel des incidents (par exemple : dépassement de seuil, panne de capteur ou de pompe) à destination des utilisateurs ou administrateurs.

3.6 Présentation de la plateforme

3.6.1 Interface utilisateur et modules fonctionnels

La plateforme Aquasmart offre une interface utilisateur moderne, épurée et responsive, facilitant la navigation et l'accès aux principales fonctionnalités. Chaque module est conçu pour répondre à un besoin précis de gestion, de supervision ou d'automatisation du système aquaponique.

1. Page d'accueil et authentification

L'utilisateur accède d'abord à une page d'accueil accueillante, suivie d'une interface de connexion sécurisée. Cette étape permet l'authentification par nom d'utilisateur ou email et mot de passe, garantissant la confidentialité et la sécurité des données.

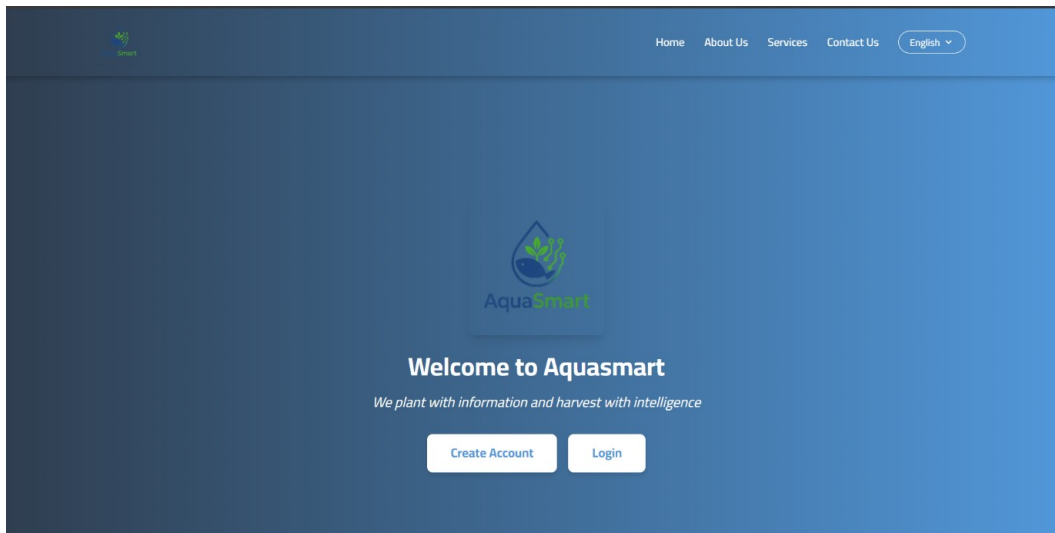


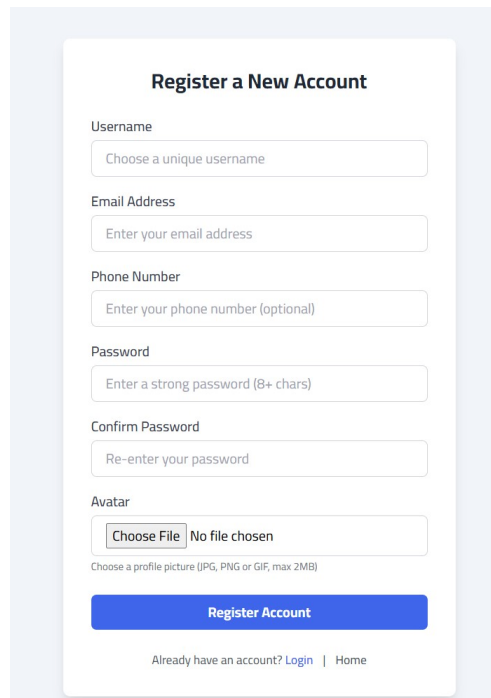
FIGURE 3.12 – Home page.

Inscription et création de compte :

La plateforme Aquasmart propose une interface intuitive pour la création de nouveaux comptes utilisateurs. Depuis la page d'inscription, l'utilisateur est invité à saisir un nom d'utilisateur unique, une adresse email valide, un numéro de téléphone (optionnel), ainsi qu'un mot de passe sécurisé et sa confirmation. Il a également la possibilité de télécharger une image de profil (avatar) pour personnaliser son compte.

Une fois le formulaire complété et validé, les informations sont enregistrées de façon sécurisée dans la base de données, avec le mot de passe chiffré. Cette étape garantit à la fois la simplicité d'accès au service pour les nouveaux utilisateurs et la sécurité des données personnelles.

Grâce à ce module, la plateforme facilite l'adoption par de nouveaux membres et assure une gestion centralisée et sécurisée des identités.



Register a New Account

Username
Choose a unique username

Email Address
Enter your email address

Phone Number
Enter your phone number (optional)

Password
Enter a strong password (8+ chars)

Confirm Password
Re-enter your password

Avatar
Choose File No file chosen
Choose a profile picture (JPG, PNG or GIF, max 2MB)

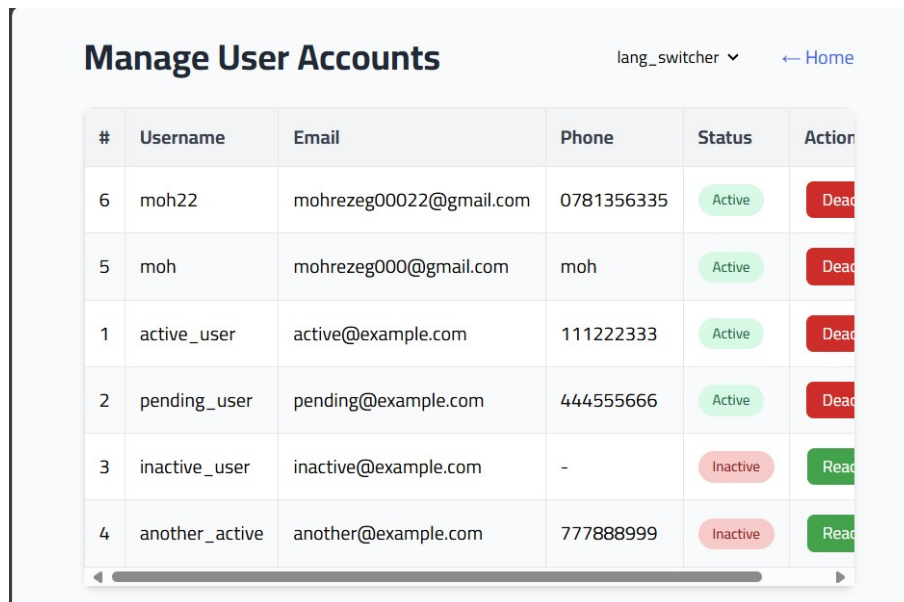
Register Account

Already have an account? [Login](#) | [Home](#)

FIGURE 3.13 – Register page.

2. Gestion des utilisateurs et administration

La plateforme Aquasmart intègre un module d'administration avancé dédié à la gestion des utilisateurs. Dès l'accès à l'application, chaque utilisateur doit s'authentifier via une interface de connexion sécurisée, en renseignant son nom d'utilisateur ou son adresse email et son mot de passe. Cette authentification garantit la confidentialité des données et permet de contrôler les accès à la plateforme selon le rôle attribué à chaque compte.



Manage User Accounts lang_switcher ▾ ← Home

#	Username	Email	Phone	Status	Action
6	moh22	mohrezeg00022@gmail.com	0781356335	Active	Deac
5	moh	mohrezeg000@gmail.com	moh	Active	Deac
1	active_user	active@example.com	111222333	Active	Deac
2	pending_user	pending@example.com	444555666	Active	Deac
3	inactive_user	inactive@example.com	-	Inactive	Reac
4	another_active	another@example.com	777888999	Inactive	Reac

FIGURE 3.14 – Page admin.

L'administrateur dispose d'une interface spécifique de gestion des comptes. Celle-ci présente une liste détaillée de tous les comptes enregistrés, incluant le nom d'utilisateur, l'email, le numéro de téléphone, le statut (actif ou inactif) ainsi que les actions possibles (activation/désactivation, suppression, réinitialisation).

Chaque utilisateur peut avoir un statut différent (actif, en attente, inactif), ce qui permet une gestion fine des droits d'accès et la sécurité du système.

Status	Actions
Active	Deactivate
Active	Deactivate
Active	Deactivate
Active	Deactivate
Inactive	Reactivate
Inactive	Reactivate

FIGURE 3.15 – Status et action des comptes.

Cette gestion centralisée facilite l'administration de la plateforme, la résolution des incidents (comme la réactivation d'un compte inactif), ainsi que la traçabilité des opérations effectuées par chaque utilisateur.

Grâce à cette approche, la solution Aquasmart garantit une gouvernance sécurisée et adaptée aux besoins des exploitants, tout en offrant une expérience utilisateur claire et professionnelle.

3. Tableau de bord (Dashboard)

catégorisation par section :

3.1 Fish (Poissons)

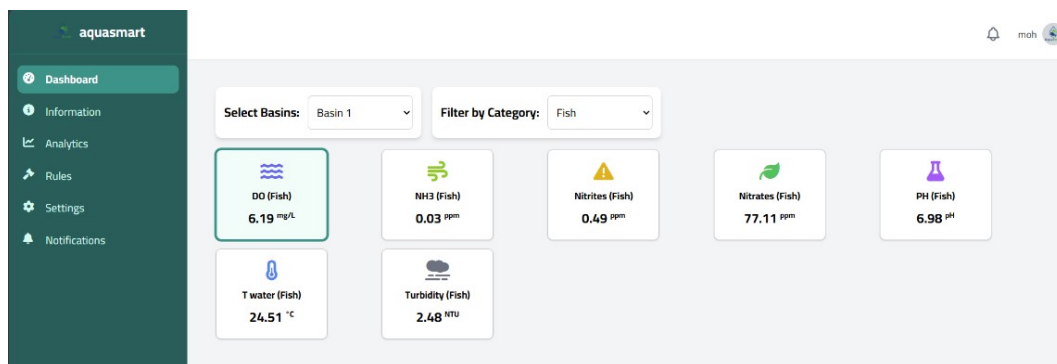


FIGURE 3.16 – Page dashboard-fish.

— **Fonctionnalité principale pour la catégorie Fish :**

Lorsque l'utilisateur sélectionne la catégorie Fish dans le tableau de bord, il accède à une vue synthétique et en temps réel des paramètres essentiels au suivi de la santé des poissons dans chaque bassin :

- Oxygène dissous (DO).
- Ammoniac (NH_3).
- Nitrites.
- Nitrates.
- pH.
- Température (T water).
- Turbidité.

— **Caractéristiques :**

- Sélection du bassin via un menu déroulant (dropdown).
- Filtrage par catégorie (Fish/Plant Section) pour adapter l'affichage aux besoins de l'utilisateur
- Affichage sous forme de cartes visuelles : chaque paramètre est représenté par une carte dédiée, affichant la valeur actuelle et l'unité correspondante.
- Interface responsive et ergonomique, facilitant la supervision rapide et la détection d'anomalies spécifiques à la section poissons.

3.2 Plant Section (Plantes)

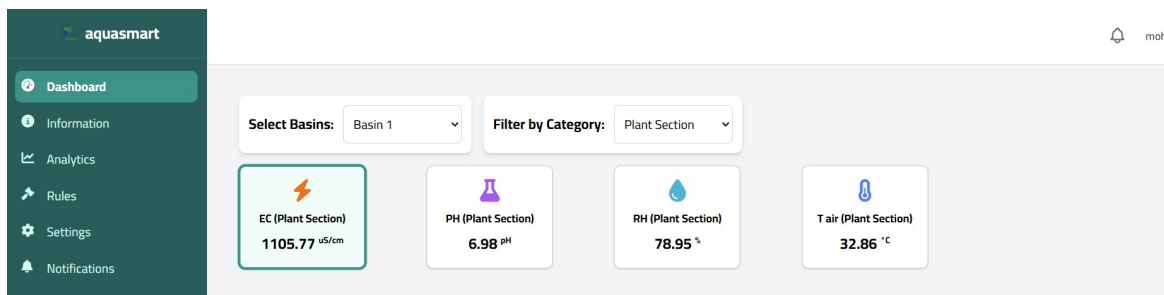


FIGURE 3.17 – Page dashboard-plant.

— **Fonctionnalité principale pour la catégorie Plant Section :**

Lorsque l'utilisateur sélectionne la catégorie Plant Section dans le tableau de bord, il visualise en temps réel les principaux paramètres liés à la zone de culture végétale du bassin, essentiels pour la croissance optimale des plantes :

- Conductivité électrique (EC).
- pH.
- Humidité relative (RH).
- Température de l'air (T air).

— **Caractéristiques :**

- Sélection du bassin via un menu déroulant.
- Filtrage par catégorie pour afficher uniquement les paramètres pertinents à la section plantes.
- Affichage sous forme de cartes visuelles : chaque paramètre apparaît dans une carte, indiquant la valeur actuelle et l'unité associée, pour une lecture rapide.
- Interface responsive et moderne, permettant une supervision claire et efficace de l'environnement des cultures végétales.

4. Module de Visualisation des Mesures (Sensor Data Visualization)

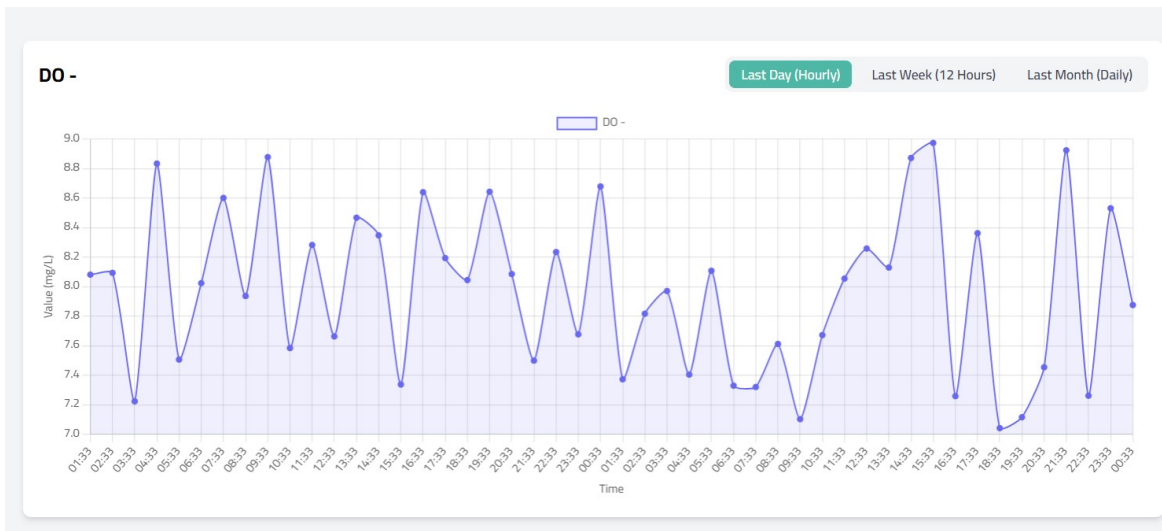


FIGURE 3.18 – Visualisation des paramètres.

Fonctionnalités principales :

— Affichage graphique des mesures :

- Courbe en ligne pour la mesure de l'oxygène dissous (DO) sur 24 heures (Last Day - Hourly).
- Axe des abscisses (temps) et axe des ordonnées (valeur mesurée).

— Navigation temporelle :

- Sélection rapide de la période d'affichage :
 - Last Day (Hourly).
 - Last Week (12 Hours).
 - Last Month (Daily).

— Visualisation des variations :

- Permet de repérer rapidement les anomalies, pics ou tendances dans la qualité de l'eau.

5. Module de Gestion des Pompes (Pumps Control Module)

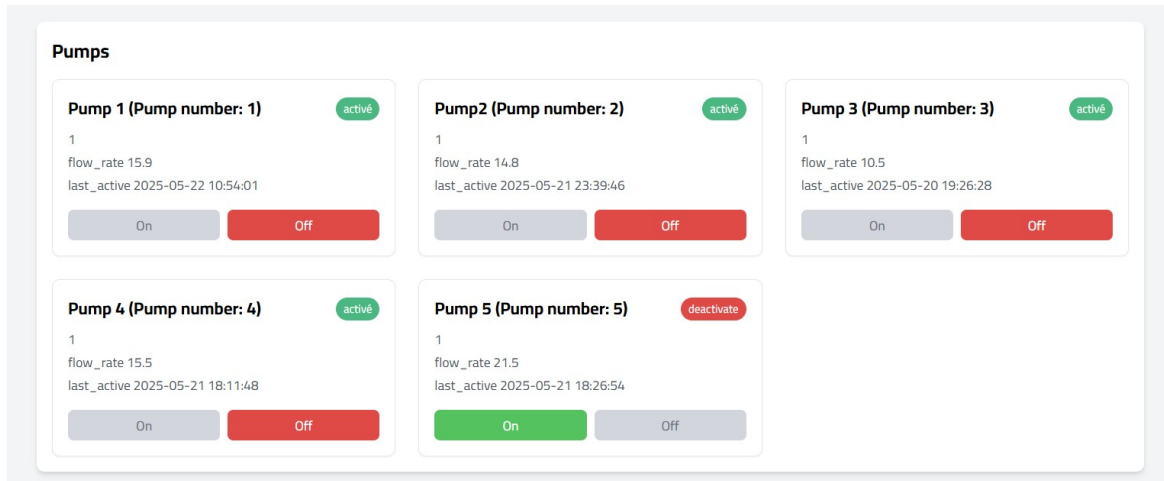


FIGURE 3.19 – Actionneur de SISCA.

Fonctionnalités principales :

- Affichage de l'état des pompes :
 - Chaque carte affiche le nom de la pompe, le numéro, le débit (`flow_rate`) et la dernière date d'activation.
- Visualisation du statut :
 - L'état (actif/inactif) est indiqué par une étiquette verte ("active") ou rouge ("deactivate").
- Contrôle manuel :
 - Deux boutons principaux pour chaque pompe :
 - On (pour activer).
 - Off (pour désactiver).
 - Les boutons sont activés/désactivés selon l'état courant de la pompe (impossible d'allumer une pompe déjà active).
- Mise à jour en temps réel :
 - Les changements de statut et d'état sont reflétés instantanément dans l'interface.

6. Information et Statistiques

Ce module permet à l'utilisateur d'accéder aux données mesurées par les différents capteurs pour chaque bassin. Il offre une interface intuitive pour sélectionner le réservoir, les capteurs souhaités et la période d'analyse. L'utilisateur peut ainsi générer des statistiques personnalisées et exporter les résultats pour une analyse approfondie.

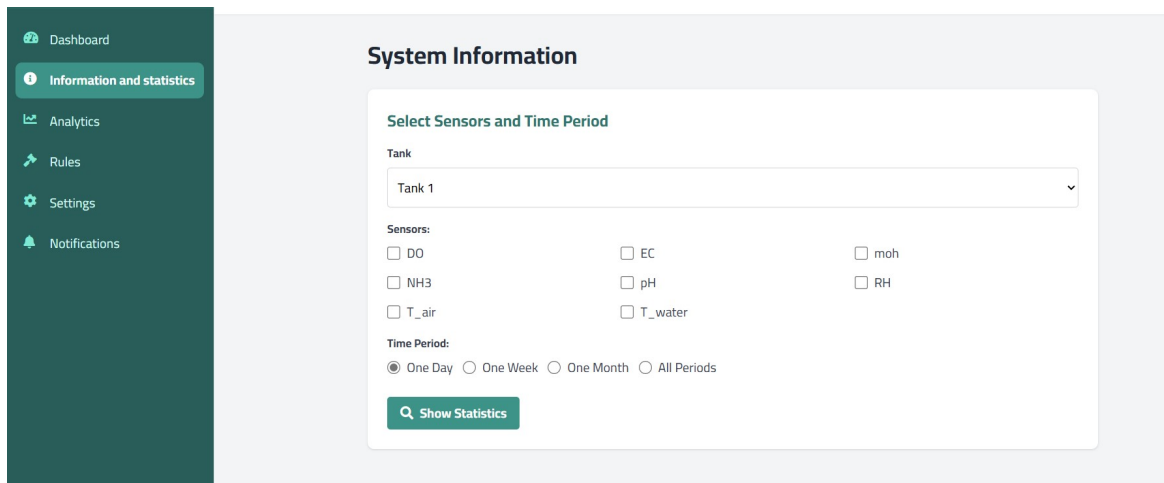


FIGURE 3.20 – Page information et statistiques.

— Fonctionnalités principales :

- **Sélection du bassin (tank) :**

L'utilisateur choisit le réservoir à analyser à partir d'un menu déroulant.

- **Choix des capteurs à analyser :**

Plusieurs capteurs sont proposés (DO, NH₃, T_air, EC, pH, T_water, RH, etc.), chaque capteur pouvant être inclus ou exclu selon le besoin.

- **Définition de la période d'analyse :**

Plusieurs options sont offertes :

- **One Day** : les dernières 24h.
- **One Week** : la dernière semaine.
- **One Month** : le dernier mois.
- **All Periods** : toute la période disponible.

- **Lancement de la génération de statistiques :**

Un bouton "Show Statistics" permet d'afficher les résultats correspondant aux paramètres sélectionnés.

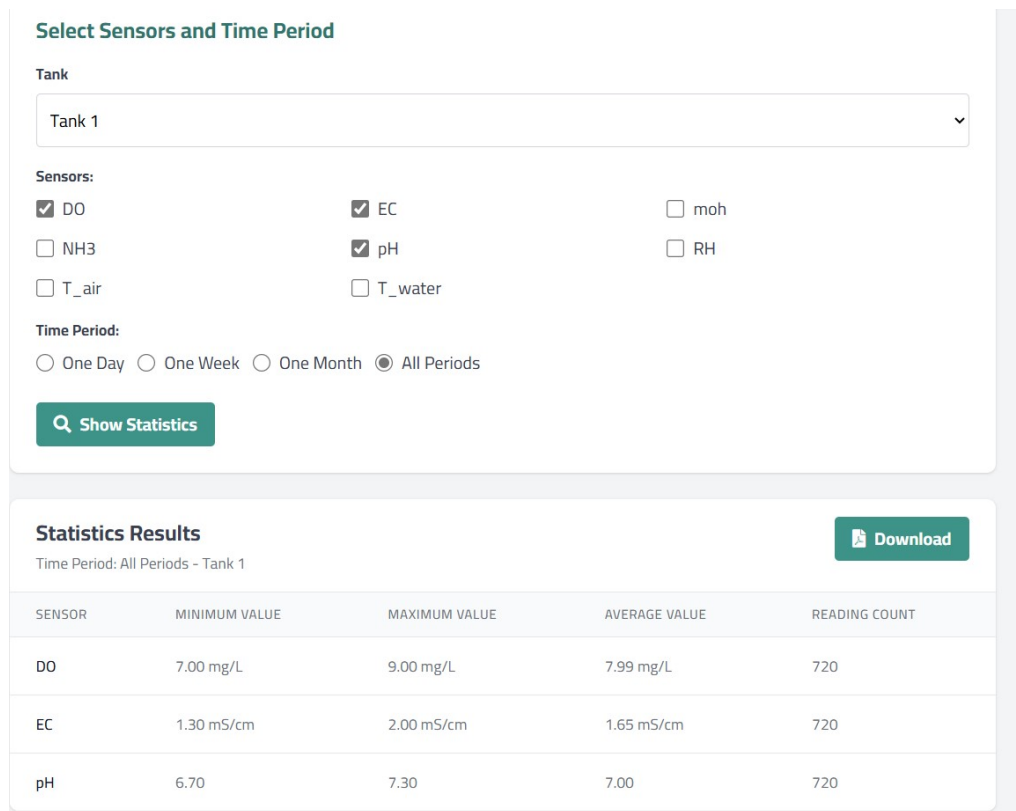


FIGURE 3.21 – Génération de statistiques.

— Génération des statistiques :

- Après avoir choisi les capteurs et la période, il suffit de cliquer sur le bouton **Show Statistics**.
- Les résultats statistiques sont affichés dans un tableau récapitulatif :
 - **Minimum Value** : Valeur minimale mesurée pour chaque capteur.
 - **Maximum Value** : Valeur maximale mesurée.
 - **Average Value** : Moyenne.
 - **Reading Count** : Nombre total de mesures prises.

— Téléchargement au format CSV :

- Un bouton **Download** permet de télécharger l'ensemble des résultats affichés sous forme d'un fichier CSV.

7. Analytique du Système

Les fonctionnalités avancées de traitement et de valorisation des données collectées dans l'aquaponie intelligente sont regroupées dans la section analytique du système.

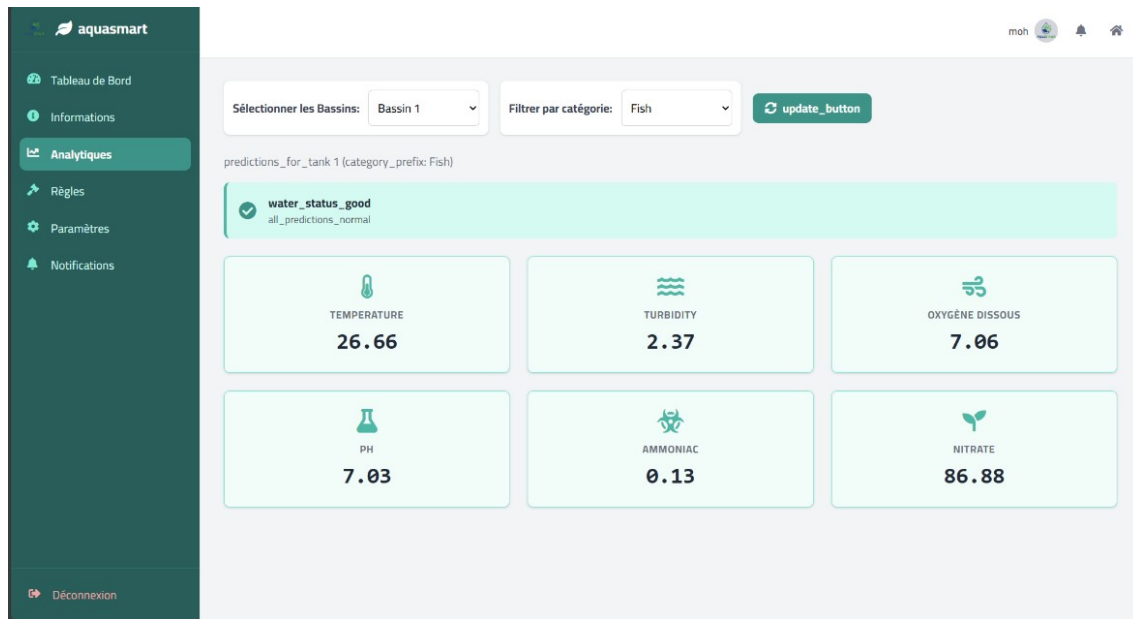


FIGURE 3.22 – Page Analytiques.

— Principales fonctionnalités :

Prédiction des paramètres importants.

Un module de prédiction affiche la valeur estimée pour l'étape suivante pour chacun des paramètres clés (température, turbidité, dissipation d'oxygène, pH, ammoniac et nitrate). Cette fonctionnalité, basée sur des modèles d'intelligence artificielle (ou d'apprentissage automatique) intégrés au système, permet d'anticiper les changements d'état du bassin et de prendre des mesures préventives.

- Bouton d'actualisation permettant de lancer une prédiction immédiate des paramètres pour la prochaine étape du système. En cliquant sur ce bouton, l'utilisateur peut visualiser non seulement l'évolution attendue pour l'étape suivante, mais également pour les étapes ultérieures (next-next step), selon les capacités du modèle prédictif.
- Si aucune anomalie n'est détectée dans les prédictions (température, pH, oxygène dissous, etc.), le système affiche automatiquement une alerte verte confirmant que toutes les valeurs sont "normales".
- Cette approche "tout va bien" permet à l'utilisateur de se concentrer sur d'autres tâches, tout en restant informé de l'état global du bassin.

— Avantages :

- Permet d'anticiper les variations à court et moyen terme.
- Offre une vision proactive pour ajuster les actions ou les réglages système avant que des dérives ne surviennent.
- Renforce la sécurité et l'optimisation du fonctionnement du bassin grâce à une gestion basée sur l'anticipation.

8. Détection des Anomalies dans les Prédictions

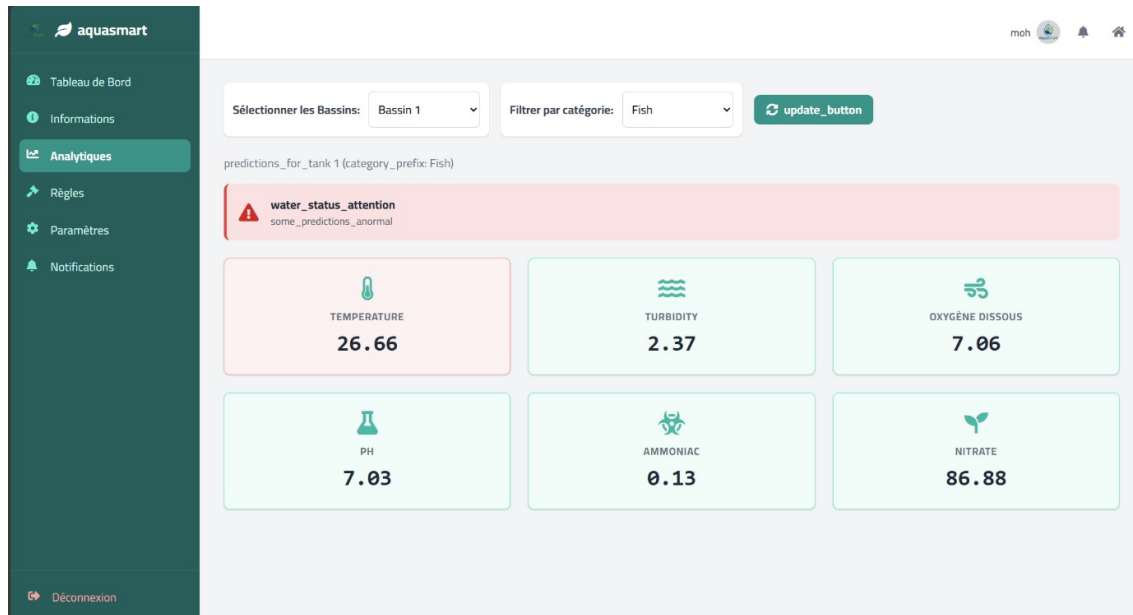


FIGURE 3.23 – Page Détection des Anomalies dans les Prédictions.

— Fonctionnement :

- Lorsqu'une prédiction dépasse les seuils définis (par exemple, température trop élevée ou taux d'ammoniac anormal), une bannière d'alerte s'affiche, indiquant au gestionnaire du bassin qu'une situation critique pourrait survenir.
- Cette alerte permet une réaction proactive : l'utilisateur est invité à vérifier l'état du bassin et à prendre les mesures nécessaires pour corriger la situation avant qu'elle ne devienne problématique.

— Avantages :

- Surveillance intelligente et en temps réel des prévisions système.
- Réduction du risque d'incident grâce à l'identification précoce des dérives.
- Interface claire et ergonomique, facilitant la prise de décision rapide.

9. Gestion des règles (Seuils des capteurs)

Le module « Rules » permet de définir, bassin par bassin, des intervalles de fonctionnement sûrs pour chaque paramètre physico-chimique et environnemental mesuré.

Select Tanks: Tank 1

sensor_thresholds

Tank 1

SENSOR TYPE	MINIMUM VALUE	MAXIMUM VALUE	UNIT
Dissolved Oxygen	0	65	mg/L
Electrical Conductivity	1000	2000	uS/cm
Ammonia	0	0.25	ppm
pH Level	6	7.5	pH
Relative Humidity	40	80	%
Air Temperature	18	35	C
Water Temperature	21.3	29.3	C

FIGURE 3.24 – Page rules.

Fonctionnalités détaillées :— **Sélection du bassin (Tank) :**

- L'administrateur ou l'utilisateur sélectionne le bassin concerné via un menu déroulant (exemple : Tank 1).
- Permet d'appliquer des règles différentes pour chaque bassin.

— **Définition des seuils pour chaque capteur :**

- Pour chaque type de capteur (Dissolved Oxygen, EC, Ammonia, pH, RH, Air Temperature, Water Temperature), l'interface affiche une ligne avec :
 - **Nom du capteur.**
 - **Valeur minimale :** à renseigner dans un champ éditable.
 - **Valeur maximale :** à renseigner dans un champ éditable.
 - **Unité de mesure** (mg/L, uS/cm, ppm, pH, %, C° ·).
- L'utilisateur ajuste ces valeurs en fonction des normes ou des besoins de l'aquaponie.

— **Interface de gestion des pompes dans le système de supervision aquaponique :**

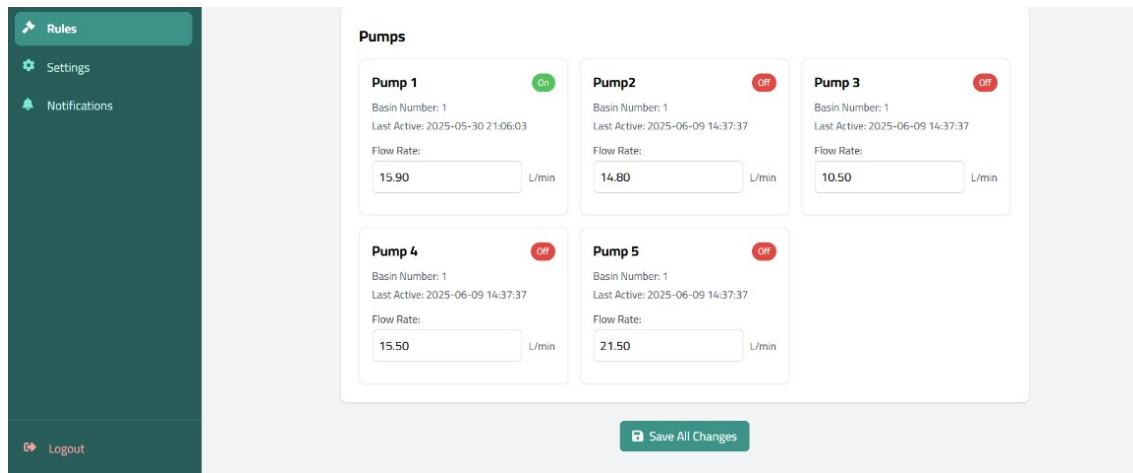


FIGURE 3.25 – Interface de gestion des pompes.

Cette interface permet de contrôler et de surveiller l'état des différentes pompes associées à un bassin spécifique (ici le bassin numéro 1). Chaque pompe dispose d'un indicateur d'état (marqué "On" en vert ou "Off" en rouge), de la date et de l'heure de sa dernière activation, ainsi que d'un champ permettant de régler le débit.

— **Enregistrement et application des règles :**

- Après modification, les seuils sont sauvegardés et appliqués automatiquement.
- Ces valeurs servent ensuite à la génération d'alertes et au contrôle automatisé du système.

10. Module « Paramètres »

Le module « Paramètres » permet à l'utilisateur de personnaliser son expérience sur la plateforme. Il regroupe différentes options de gestion de compte et de configuration de l'application, accessibles à tous les utilisateurs disposant d'un compte.

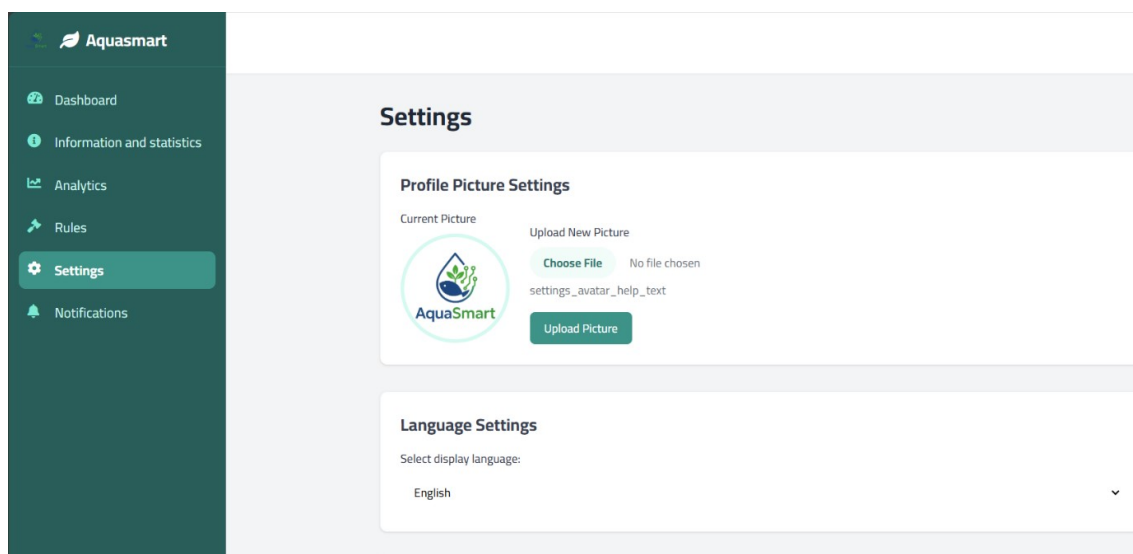


FIGURE 3.26 – Page settings.

— **Fonctionnalités principales :**

• **Modification de l'avatar (photo de profil) :**

- L'utilisateur peut consulter sa photo de profil actuelle.
- Il a la possibilité d'importer une nouvelle image (formats JPG, PNG, GIF, taille maximale 2 Mo).
- Un bouton « Upload Picture » permet d'enregistrer la nouvelle image sur le serveur.

• **Changement de la langue d'affichage :**

- Un menu déroulant permet de sélectionner la langue d'affichage de l'interface (par exemple, Anglais ou Français).
- Ce choix est enregistré et appliqué instantanément à l'ensemble de l'application.

Arabe :

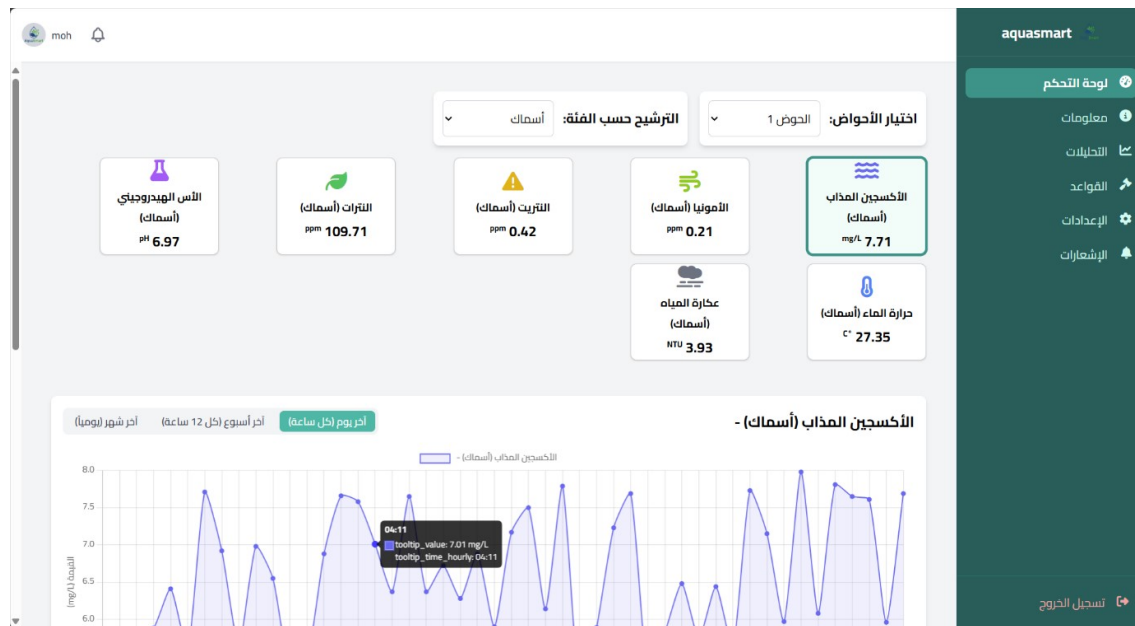


FIGURE 3.27 – Page en arabe.

— **Gestion du mot de passe :**

- L'utilisateur peut modifier son mot de passe depuis l'interface.
- Il doit renseigner son mot de passe actuel, puis saisir et confirmer un nouveau mot de passe (respectant la politique de sécurité, ex : minimum 8 caractères).
- Un bouton « update_password » permet de valider la modification.

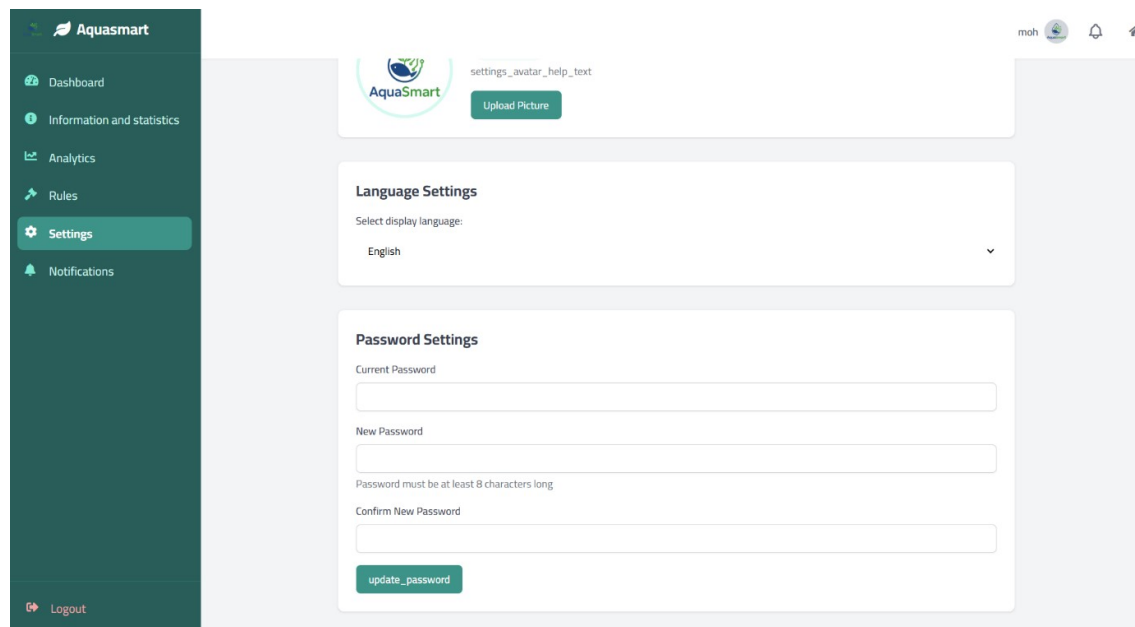


FIGURE 3.28 – Gestion du mot de passe.

11. Module « Notifications »

Le module « Notifications » a pour objectif d'alerter l'utilisateur en temps réel lorsqu'un paramètre critique du système sort de sa plage de sécurité ou lorsqu'un événement important se produit (alerte de seuil, panne, anomalie...).

Toutes les notifications générées sont centralisées dans ce module et affichées sous forme de liste, accessible via le menu latéral.

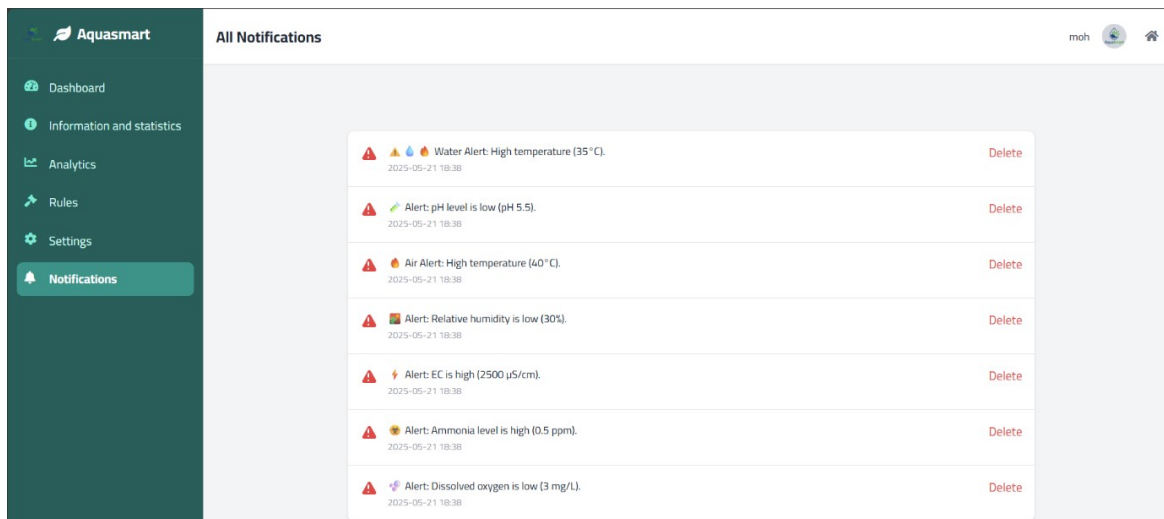


FIGURE 3.29 – Page de notifications.

Fonctionnalités principales :

— Affichage des notifications :

- Toutes les alertes sont listées de manière chronologique, les plus récentes en haut.

- Chaque notification comporte :
 - Un titre et un message explicite (ex : « Water Alert : High temperature (35°C) », « Alert : pH level is low (pH 5.5) » ...),
 - La date et l'heure précise de l'événement.
- **Suppression d'une notification :**
 - Un bouton « Delete » permet à l'utilisateur de supprimer une notification de la liste une fois l'alerte traitée.
 - Cette suppression permet de garder une interface claire et de se concentrer sur les alertes en cours.
- **Gestion multi-paramètres :**
 - Le système peut générer des notifications pour différents types de paramètres surveillés : température de l'eau, pH, ammoniac, humidité relative, conductivité électrique, oxygène dissous, etc.

3.7 Conclusion

Ce chapitre a permis de détailler l'ensemble du processus d'implémentation du système SISCA, de la configuration des environnements techniques à l'intégration des modules logiciels et matériels. Grâce à une architecture robuste et modulaire, combinant des technologies web modernes, une base de données relationnelle optimisée, un firmware embarqué en C++ et un modèle prédictif basé sur le Deep Learning (GRU et Random Forest), la plateforme Aquasmart répond efficacement aux exigences d'un système de supervision aquaponique intelligent.

L'intégration réussie des capteurs IoT, des actionneurs contrôlables à distance, des modules d'alerte, de visualisation de données et de prédiction automatique confère au système une dimension proactive et autonome. Le tableau de bord interactif offre une expérience utilisateur fluide, tandis que les résultats expérimentaux démontrent la capacité du modèle à anticiper l'évolution de paramètres critiques avec un bon niveau de précision, hormis quelques limites sur certaines variables comme l'ammoniaque.

Ainsi, cette phase d'implémentation représente une étape clé de la réalisation du projet, posant les bases solides pour les phases ultérieures d'optimisation, de déploiement à plus grande échelle, et d'intégration de fonctionnalités supplémentaires.

Conclusion Générale

À la suite de ce projet, nous avons développé et mis en œuvre SISCA (Système intelligent de surveillance et de contrôle pour l'aquaponie), une solution innovante combinant cloud computing, intelligence artificielle et Internet des objets (IoT) pour optimiser la gestion des fermes aquaponiques. Ce système permet de recueillir en temps réel des données importantes sur l'environnement aquatique, notamment la température, le pH, la dissipation d'oxygène, l'ammoniac, les nitrates, etc. Ces données sont ensuite transmises via le protocole MQTT vers un serveur cloud, où elles sont stockées, visualisées et analysées. L'intégration de modèles d'IA, tels que le GRU, a permis de démontrer la capacité du système à classifier les états de l'eau, détecter des anomalies et prédire les dérives environnementales. De plus, il offre une interface web intuitive permettant le contrôle, le diagnostic et la visualisation des paramètres à distance.

La méthode choisie repose sur une architecture modulaire composée d'une unité interne (serveur cloud, base de données, interface utilisateur et module d'IA) et d'une unité externe (capteurs, microcontrôleurs et actionneurs). De plus, le système a été conçu selon une logique SaaS (Software as a Service), permettant une gestion multi-utilisateurs, sécurisée et personnalisable, et mettant l'innovation à la disposition de divers exploitants agricoles.

En revanche, l'intégration plus sophistiquée de l'intelligence artificielle ouvre de nouvelles possibilités d'anticipation des déséquilibres, d'optimisation de la nutrition des plantes et des poissons, et d'automatisation de la prise de décision. Cependant, ce travail n'est qu'une première étape. Plusieurs perspectives d'amélioration sont envisagées :

Intégration de techniques de deep learning plus avancées pour des prédictions long terme plus fiables. À long terme, cette approche pourrait se traduire par une augmentation significative de la productivité, une réduction des pertes et une gestion plus durable des ressources, en particulier dans un contexte local comme celui de l'Algérie, où les enjeux de sécurité hydrique et alimentaire constituent une priorité nationale. Ainsi, SISCA représente un pas important vers l'agriculture intelligente, alliant technologie, durabilité et accessibilité, tout en posant les bases d'évolutions futures prometteuses dans le domaine de l'aquaponie.

Bibliographie

- [1] K. N. AZAD et M. A. SALAM. “Aquaponics in Bangladesh : Current status and future prospects”. In : *Journal of Bioscience and Agricultural Research* 7.2 (2016), p. 669-677.
- [2] N. MCHUNU, G. LAGERWALL et A. SENZANJE. “Aquaponics in South Africa : Results of a national survey”. In : *Aquaculture Reports* 12 (nov. 2018), p. 12-19.
- [3] AQUAPONIA. *Qu’est-ce que l’Aquaponie : Définition*. 2012. URL : <https://www.aquaponia.com/aquaponie-by-echologia/monde-aquaponia/qu-est-ce-que-l-aquaponie-definition>.
- [4] PIXABAY. *Pill Icon Vector – Image free from Pixabay*. <https://pixabay.com/vectors/pill/>. 2025.
- [5] YOU MATTER. *What Is The Aquaponics System ? Definition, Benefits, Weaknesses*. 2024. URL : <https://youmatter.world/en/definition/aquaponics-sustainable-benefits-system/>.
- [6] Schuyler ‘Rocky’ REIDEL. *Aquaponics Water Quality Parameters Cheatsheet*. 2023. URL : <https://friendlyaquaponics.com/aquaponics-water-quality-parameters-cheatsheet/>.
- [7] Live to PLANT. *How to Maintain Water Quality in Aquaponics Systems*. 2025. URL : <https://livetoplant.com/how-to-maintain-water-quality-in-aquaponics-systems/>.
- [8] Ph.D. JASON DANAHER. *What Are the Basic Water Quality Parameters to Monitor for Aquaponic Systems*. 2025. URL : <https://www.aquaticed.com/blogs/deeper-dive/what-are-the-basic-water-quality-parameters-to-monitor-for-aquaponic-systems>.
- [9] GO GREEN AQUAPONICS. *How to Test Water Quality in Aquaponics*. 2025. URL : <https://gogreenaquaponics.com/blogs/news/how-to-test-water-quality-in-aquaponics>.
- [10] Thibaut GILLES. *Internet des objets (IoT) : au cœur de l’agriculture 4.0*. 2021. URL : <https://www.blog-qhse.com/internet-des-objets-iot-au-coeur-de-lagriculture-4.0>.
- [11] Dayu WANG, Daojun ZHONG et Alireza SOURI. “Energy management solutions in the Internet of Things applications : Technical analysis and new research directions”. In : *Cognitive Systems Research* 67 (2021), p. 33-49. ISSN : 1389-0417. DOI : <https://doi.org/10.1016/j.cogsys.2020.12.009>. URL : <https://www.sciencedirect.com/science/article/pii/S1389041720301157>.
- [12] Ovidiu VERMESAN et Peter FRIESS. *Internet of Things Applications–From Research and Innovation to Market Deployment*. Boca Raton, FL : Taylor & Francis, 2014. ISBN : 978-1-4822-2714-7.
- [13] DIGI INTERNATIONAL. *L’IoT dans l’agriculture : 10 cas d’utilisation des technologies agricoles intelligentes*. 2025. URL : <https://fr.digi.com/blog/post/iot-in-agriculture>.

- [14] WIRELESS LOGIC. *IoT pour l'agriculture*. 2025. URL : <https://wirelesslogic.com/fr/secteurs/agriculture>.
- [15] FNB TECH. *L'Internet des objets dans l'agriculture : Optimisation des ressources agricoles*. 2025. URL : <https://fnb.tech/fr/iot-in-agriculture/>.
- [16] Mary K. PRATT. *Top 12 most commonly used IoT protocols and standards*. Mai 2025. URL : <https://www.techtarget.com/iotagenda/tip/Top-12-most-commonly-used-IoT-protocols-and-standards>.
- [17] COM4. *An In-Depth Guide to IoT Protocols*. Juin 2025. URL : <https://www.com4.no/en/blog/an-in-depth-guide-to-iot-protocols>.
- [18] Paessler AG. *MQTT, qu'est-ce que c'est ? Définition et explications détaillées*. 2025. URL : <https://www.paessler.com/fr/it-explained/mqtt>.
- [19] AMAZON WEB SERVICES. *Qu'est-ce que le protocole MQTT ?* 2025. URL : <https://aws.amazon.com/fr/what-is/mqtt/>.
- [20] COMMISSION NATIONALE DE L'INFORMATIQUE ET DES LIBERTÉS (CNIL). *Intelligence artificielle*. 2025. URL : <https://www.cnil.fr/fr/definition/intelligence-artificielle>.
- [21] BIENVENUM. *Comment l'IA révolutionne l'agriculture moderne*. 2025. URL : <https://bienvenum.org/comment-lia-revolutionne-lagriculture-moderne/>.
- [22] IA SCHOOL. *Comment l'intelligence artificielle transforme l'agriculture ?* 2025. URL : <https://www.intelligence-artificielle-school.com/alternance-et-entreprises/secteur-d-activite/comment-lintelligence-artificielle-transforme-lagriculture/>.
- [23] Mohammad HARIS et Rafiqul Zaman KHAN. "A Systematic Review on Cloud Computing". In : *International Journal of Computer Sciences and Engineering* 6.11 (2018), p. 632-639.
- [24] HEWLETT PACKARD ENTERPRISE. *What is Cloud Computing ?* 2025. URL : https://www.hpe.com/emea_europe/en/what-is/cloud-computing.html#:~:text=Cloud%20computing%20is%20an%20approach,and%20other%20resources%20on%20demand..
- [25] Aaqib RASHID et Amit CHATURVEDI. "Cloud Computing Characteristics and Services : A Brief Review". In : *International Journal of Computer Sciences and Engineering* 7.2 (2019), p. 421-426.
- [26] Mohammad Ilyas MALIK, Shahid Hussain WANI et Abreen RASHID. "Cloud Computing–Technologies". In : *International Journal of Advanced Research in Computer Science* 9.2 (2018).
- [27] Shanmugasundaram MUNIANDI. *Reference Model of Cloud Computing : SaaS Provides Application-Based Service on Demand*. 2017. URL : https://www.researchgate.net/figure/Reference-model-of-cloud-computing-SaaS-provides-application-based-service-on-demand_fig1_322347800.
- [28] ORACLE CORPORATION. *Qu'est-ce qu'un SaaS ?* 2025. URL : <https://www.oracle.com/fr/cloud/definition-saas/>.
- [29] ESPARKBIZ TECHNOLOGIES. *Architecture SaaS : Conception et Déploiement*. 2025. URL : <https://www.esparkinfo.com/software-development/saas/architecture>.

- [30] Priyanshu SRIVASTAVA et Rizwan KHAN. “A Review Paper on Cloud Computing”. In : *International Journal of Advanced Research in Computer Science and Software Engineering* 8.6 (2018), p. 17-20.
- [31] Alessio BOTTA et al. “Integration of Cloud Computing and Internet of Things : A Survey”. In : *Future Generation Computer Systems* 56 (2016), p. 684-700. DOI : [10.1016/j.future.2015.09.021](https://doi.org/10.1016/j.future.2015.09.021).
- [32] B. NEMADE et D. SHAH. “An IoT-Based Efficient Water Quality Prediction System for Aquaponics Farming”. In : *Computational Intelligence*. Springer, 2023, p. 311-323.
- [33] M. M. ISLAM et al. “Monitoring Water Quality Metrics of Ponds with IoT Sensors and Machine Learning to Predict Fish Species Survival”. In : *Microprocessors and Microsystems* 102 (2023), p. 104930. DOI : [10.1016/j.micpro.2023.104930](https://doi.org/10.1016/j.micpro.2023.104930).
- [34] R. E. RODRIGUEZ et al. “Prediction Model Based on Bagging and Boosting Ensemble Technique for Decision Support System of Autonomous Smart IIoT Smart Aquaponic System”. In : *Journal of Physics : Conference Series*. T. 2559. 1. IOP Publishing, 2023, p. 012015. DOI : [10.1088/1742-6596/2559/1/012015](https://doi.org/10.1088/1742-6596/2559/1/012015).
- [35] D. KARIMANZIRA et T. RAUSCHENBACH. “Enhancing Aquaponics Management with IoT-based Predictive Analytics for Efficient Information Utilization”. In : *Information Processing in Agriculture* 6.3 (2019), p. 375-385. DOI : [10.1016/j.inpa.2019.01.002](https://doi.org/10.1016/j.inpa.2019.01.002).
- [36] MDN WEB DOCS. *Web Technology for Developers*. English. 2022. URL : <https://developer.mozilla.org/en-US/docs/Web>.
- [37] Fernando SOBRAL. *Tailwind CSS v4 : What Developers Need to Know*. Mars 2025. URL : <https://eagerworks.com/blog/tailwind-css-v4>.
- [38] Bastien L. *Qu’est-ce que SQL ? Tout savoir sur le langage des bases de données*. fr-FR. 2022. URL : <https://www.lebigdata.fr/sql-tout-savoir-guide>.
- [39] PCMAG. *Definition of PHP*. en-ENG. 2022. URL : <https://www.pcmag.com/encyclopedia/term/php>.
- [40] Guido van ROSSUM. *What is Python ? Executive Summary*. <https://www.python.org/doc/essays/blurb/>. 1999. URL : <https://www.python.org/doc/essays/blurb/>.
- [41] COMMISSION NATIONALE DE L’INFORMATIQUE ET DES LIBERTÉS (CNIL). *Interface de programmation d’application (API)*. fr-FR. 2022. URL : <https://www.cnil.fr/fr/definition/interface-de-programmation-dapplication-api>.
- [42] ARDUINO. *Arduino IDE v1 Basics*. en-ENG. 2023. URL : <https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics>.
- [43] MICROSOFT. *Documentation for Visual Studio Code*. en-ENG. 2022. URL : <https://code.visualstudio.com/docs>.
- [44] Jérémy ROBERT. *Deep Learning ou Apprentissage Profond : qu’est-ce que c’est ?* français. 2020. URL : <https://datascientest.com/deep-learning-definition>.

- [45] Intelligence Artificielle SCHOOL. *Qu'est-ce qu'une Gated Recurrent Unit (GRU) ?* français. 2025. URL : <https://www.intelligence-artificielle-school.com/ecole/technologies/gated-recurrent-unit/>.
- [46] Devoteam FRANCE. *Algorithme N°2 – Comprendre comment fonctionne un random forest en 5 min.* français. 2025. URL : <https://france.devoteam.com/paroles-dexperts/algorithme-n2-comprendre-comment-fonctionne-un-random-forest-en-5-min/>.