

**Faculté des Sciences Exactes et d'Informatique**  
**Département de Mathématiques et informatique**  
**Filière : Informatique**

## RAPPORT DE PROJET DE MASTER EN INFORMATIQUE

Option : **Ingénierie des Systèmes d'Information**

THEME :

### ÉVALUATION DE LA PERFORMANCE ET DE STOCKAGE DANS LES LABORATOIRES DISTANTS

Etudiant(e) : **Benzaza Senoucia**

Etudiant(e) : **Benchehida Bouchra Nour Elhouda**

Encadrant(e) : **Moussa Mohamed**

Année Universitaire 2022-2023

---

# Remerciements

---

Nous souhaitons exprimer notre profonde gratitude envers le Bon Dieu pour nous avoir accordé le courage et la patience nécessaire pour mener ce projet à terme.

Nos remerciements les plus sincères vont à notre enseignant et promoteur, Mr. MOUSSA, pour avoir accepté de nous encadrer et de nous proposer ce sujet. Nous sommes reconnaissantes pour son aide précieuse, son travail encourageant et ses conseils avisés tout au long de notre parcours.

Nous tenons également à remercier les membres du jury d'avoir accepté d'évaluer notre travail. Leur participation et leur expertise ont contribué à l'enrichissement de notre projet. Nous n'oublions pas de remercier tous les enseignants et étudiants du département d'informatique de l'FSEI pour leur soutien et leur collaboration.

Nos familles, nos amis et toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce travail méritent également nos remerciements sincères. Leur soutien inconditionnel et leurs encouragements ont été essentiels pour notre réussite.

---

---

"if you can't come to the lab, the lab will come to you"

---

Jesus Del Alamo

## - الملخص

يتمثل عملنا في تقديم نهج لتخزين بيانات القياس للمختبرات عن بعد. بعد ذلك يتركز العمل حول تقبيل محركات التخزين (DBMS): MongoDB و Postgres ، فيما يتعلق بإستراتيجيات التحجيم الأفقي الخاصة بكل منهما. سيعتمد التنفيذ على نهج خدمة المنحى .

## الكلمات مفتاحية

معمل بعيد - قاعدة بيانات - اختبار تحميل - مستخدمين - أداء

## - Résumé :

Notre travail consiste à présenter une approche pour le stockage des données de mesures pour les laboratoires distants. Ensuite, le travail se concentre sur l'évaluation des moteurs de stockage, en exprimant deux types des systèmes de gestion de base de données open source (SGBD); MongoDB et PostgreSQL, par rapport à leurs stratégies de mise à l'échelle horizontale respective. L'implémentation sera basée sur une approche orientée services.

## mots clés

Laboratoire à distance - Base de données - Test de chargement - Utilisateurs - Performances

## - Abstract

Our work is to present an approach for storing measurement data in remote laboratories. Next, the work focuses on evaluating two types of open source database management systems (DBMS); MongoDB and PostgreSQL, with respect to their respective horizontal scaling strategies. The implementation will be based on a service-oriented approach.

## key words

Remote lab - Database - load test - Users - Performance

# Table des matières

<b>1</b>	<b>État de l'art</b>	<b>2</b>
1.1	Introduction : . . . . .	3
1.2	Définition des laboratoires distants : . . . . .	3
1.3	Historique des laboratoires distants : . . . . .	4
1.4	L'intérêt des laboratoires distants : . . . . .	4
1.5	Composition des laboratoires distants : . . . . .	4
1.5.1	Matériel : . . . . .	5
1.5.2	Logiciel : . . . . .	5
1.5.3	Communications : . . . . .	6
1.6	Modes de travaux de laboratoire : . . . . .	7
1.6.1	Laboratoire pratique conventionnel : . . . . .	7
1.6.2	Laboratoire non-conventionnel : . . . . .	8
1.6.3	Avantages des modes de travaux de laboratoire : . . . . .	8
1.6.4	LAB pratique "traditionnel" : . . . . .	8
1.6.5	LAB à distance : . . . . .	8
1.6.6	LAB virtuel : . . . . .	8
1.7	Les apprentissages réalisés par les travaux de laboratoire : . . . . .	9
1.8	Les types de laboratoires : . . . . .	11
1.9	Les défis des laboratoires distants : . . . . .	12
1.10	Système de gestion des laboratoires distants : . . . . .	12
1.11	Les fonctionnalités du RLMS : . . . . .	13
1.12	Implémentation des RLMS : . . . . .	13
1.13	Partage des laboratoires distants : . . . . .	14
1.14	Conclusion : . . . . .	14
<b>2</b>	<b>Évaluation des performances</b>	<b>15</b>
2.1	Introduction : . . . . .	16
2.2	Les étapes d'un LAD (laboratoire à distance) d'un étudiant : . . . . .	16
2.2.1	Pré-lad : . . . . .	16
2.2.2	Lad : . . . . .	16
2.2.3	Post-lad : . . . . .	16

2.2.4	LAD en cas d'impossibilité d'accès aux équipements par l'étudiant : . . . . .	16
2.3	Quelques travaux de LAD des pays du monde : . . . . .	17
2.3.1	Lab@home : . . . . .	17
2.3.2	L'innovation à l'UQO :(Université Québec en Outaouais) . . .	17
2.3.3	Projet laboratoire à distance LAD (Remote lab) à Montréal : .	22
2.3.4	Ilab Shared Architecture : . . . . .	22
2.3.5	RemLabNet : . . . . .	24
2.3.6	Weblab : Université de Deusto en Espagne . . . . .	25
2.3.7	VISIR (Virtual Instrument Systems in Reality) : Blekinge Institute of Technology (BTH) in Sweden 2015 : . . . . .	26
2.4	Les travaux de LAD en Algérie : . . . . .	27
2.5	MostaLab -Mostaganem- : . . . . .	27
2.5.1	Contraintes fonctionelles de Mostalab : . . . . .	27
2.5.2	Contraintes non-fonctionelles : "techniques" . . . . .	27
2.5.3	Besoins fonctionelles : . . . . .	28
2.6	Vue sur quelques fameux laboratoire virtuel : MSOL, LABERSIME : . .	28
2.6.1	1-MSOL (Massively scalable online laboratories) . . . . .	28
2.6.2	2- LABERSIME (École polytechnique d'Agadir-Maroc-) : . . . .	29
2.7	SQL et NOSQL : . . . . .	30
2.7.1	La base de données SQL : . . . . .	30
2.7.2	Base de données NOSQL : . . . . .	30
2.8	Comparaison entre SQL et NOSQL : . . . . .	31
2.9	Conclusion : . . . . .	32
<b>3</b>	<b>Étude comparatif de la performance de R-LAB</b>	<b>33</b>
3.1	Introduction : . . . . .	34
3.2	Les outils utilisés : . . . . .	34
3.2.1	Visual Studio Code : . . . . .	34
3.2.2	Fonctionnalités de Visual Studio Code : . . . . .	34
3.2.3	Overleaf : . . . . .	34
3.2.4	Latex : . . . . .	34
3.2.5	Fonctionnalités du Latex : . . . . .	34
3.2.6	Artillery.io : . . . . .	34
3.2.7	Installation et lancement de l'outil Artillery.io : . . . . .	35
3.2.8	DBeaver : . . . . .	37
3.2.9	MongoDB : . . . . .	38
3.3	Le Big Data : . . . . .	38
3.4	Langages, Frameworks, Plateformes : . . . . .	39
3.4.1	API : . . . . .	39
3.4.2	JavaScript : . . . . .	40

3.4.3	YAML :	40
3.4.4	Node js :	41
3.4.5	Nodemon :	41
3.4.6	Express js :	41
3.5	Évaluation de la performance dans l'environnement NoSql et SQL :	43
3.5.1	1- Dans l'environnement NoSQL :	43
3.5.2	2- Dans l'environnement SQL :	45
3.6	Conclusion :	49
3.6.1	Thèse Doctorat :	56
3.6.2	Google Scholar :	56
3.6.3	Youtube :	56
3.6.4	PDF :	57
3.6.5	Article :	57
3.6.6	website :	57

# Table des figures

Figure	Signification	N
Figure1	Evaluation de l'internet depuis 90s	4
Figure2	Oscilloscope	5
Figure3	Cycle de vie de l'application web	6
Figure4	fonctionnalités de l'IOT	6
Figure5	La composition d'un laboratoire distant	7
Figure6	Laboratoire traditionnel	7
Figure7	exemple d'un laboratoire virtuel	8
Figure8	étude faite sur les travaux conventionnels et non conventionnels	10
Figure9	ensemble d'analyse des critères d'apprentissages	11
Figure10	image représentant un laboratoire en science	12
Figure11	implémentation RLMS	14
Figure12	image représentant un logiciel qui remplace différents instruments électroniques	17
Figure13	réalisation d'un laboratoire distant	18
Figure14	dispositif de manipulation	18
Figure15	LAD de l'UQO	19
Figure16	La session Zoom réalisé par l'enseignant	20
Figure17	deuxième partie de la manipulation	20
Figure18	Soutien de l'enseignant en rejoignant les salles des étudiants	21
Figure19	Remote lab à Montréal	22
Figure20	topologie architecturale de iLab	24
Figure21	Fonctions générales de RemLabNet	25
Figure22	l'architecture du weblab	26

Continued on next page

Table 0: (Continued)

Figure23	interface de Visir	27
Figure24	Exemple d'un déploiement de MostaLab	28
Figure25	LABERSIME interactions	30
Figure26	SQL Vs NoSQL	32
Figure27	Table SQL crée en DBeaver	39
Figure28	Table SQL créer en DBeaver	40
Figure29	Table NoSQL créer en MongoDB Compass	40
Figure30	interactions de l'API	41
Figure31	Node.js architecture	44
Figure32	Node HTTP serveur	45
Figure33	l'ensemble des phases de Artillery.io	51

## Liste des tableaux

Tableau	Description	Page
Tableau01	Regard comparatif des trois modes	8
Tableau02	Tableau d'évaluation des performances pour atteindre un temps de réponse de 5s dans NoSQL	40
Tableau03	Tableau d'évaluation des performances pour atteindre un temps de réponse de 5s dans SQL	42
Tableau04	Tableau d'évaluation des performances pour atteindre un temps de réponse de 5s dans SQL avec 5M de data	43

## Table des abréviations

<b>l'acronyme</b>	<b>Signification</b>
RLMS	Remote Laboratories Management System
IOS	Internet Of Objects
LAB	Laboratoire
LAD	laboratoire à distance
UQO	Université Québec en Outaouais
SOAP	Simple Object Access Protocol
WSDL	Web Services Description Language
Moodle	Modular Object-Oriented Dynamic Learning Environment
SQL	Structured Query Language
CRUD	Create , Read , Update , Delete
HTTP	Hypertext Transfer Protocol
API	Application Programming Interface

# Introduction Générale

## Introduction :

Dans un contexte éducatif, les travaux pratiques occupent un rôle crucial pour le processus d'apprentissage d'une vie universitaire d'un étudiant en ingénierie, ou bien en sciences appliquées. Le progrès technologique dans les ordinateurs, les réseaux et les communications ont donné naissance à de nouvelles méthodes expérimentales telles que les expériences virtuelles et à distance. D'ailleurs les laboratoires en ligne forment une solution pour certaines institutions dont ils forment leurs étudiants à distance ou pour certains laboratoires mal équipés ou bien n'ont même pas de laboratoires. Cela a obligé la migration du laboratoire traditionnel vers le laboratoire distant, réduisant tel problèmes comme le budget, les équipements, la maintenance...etc.

Un laboratoire à distance offre l'accès à distance via le réseau internet à des équipements réels et des expérimentations potentielles situé dans un tel endroit. Avec le cursus temporelle et l'amélioration des laboratoires à distance, des efforts ont été fournis pour répondre à un grand nombre d'utilisateurs.

## Organisation du travail :

On a débuter notre travail en 1 er chapitre par citer les notions les plus importantes sur ce qui concerne les laboratoires distants(une vue globale sur les définitions, historique, les défis ...) et certains notions (RLMS), en 2 eme chapitre on a mis en évidence les laboratoires distants trouvés en monde et leur fonctionnements, une comparaison (SQL et Nosql), et pour en conclure, le 3eme chapitre, nous avons défini une méthodologie d'évaluation des performances entre les bases de données SQL et NoSQL. Pour cela on a utilisé Artillery, un outil qui permet de simuler des utilisateurs virtuels afin de faciliter notre travail.

# Chapitre 1

## État de l'art

## 1.1 Introduction :

L'arrivée du covid-19 met sur table la nécessité de tout faire à distance incluant tout sorte de secteur. En université, au secteur des enseignements scientifiques et techniques, Les travaux en laboratoire représentent des activités très demandâtes pour former l'étudiant en séances des travaux pratiques, cela est pareil à leurs enseignants afin de réussir leurs études supérieures, ainsi leurs recherches. Cela a fait appel à la notion de notre premier chapitre « les laboratoires distants ».

## 1.2 Définition des laboratoires distants :

En littérature scientifique, il existe plusieurs définitions de ce qu'il s'agit d'un laboratoire distant et ses caractéristiques, par conséquent il n'existe pas une description universelle acceptée.

Par exemple, " Chen et Al " on fournit la définition suivante :  
Un laboratoire distant est une expérience contrôlée à distance via le réseau mondiale "Internet", avec l'utilisation des instruments réelles dans un endroit différent de celui où ils sont contrôlés.[31]

"Umborg et Uukkivi" on définit les laboratoires à distance comme ceci :  
Les laboratoires distants utilisent les moyens de télécommunications pour réaliser des expériences de laboratoire en présence des objets réels et des outils de mesures, afin d'atteindre un but précis.[32]

Un laboratoire distant est un laboratoire virtuel qui donne accès aux équipements réels via le navigateur web, ce laboratoire permet d'envoyer des commandes de contrôles pré-traités afin de ne pas risquer l'équipement réel cher, et d'exécuter les expériences sur l'équipement réel dans un vrai laboratoire. En dernier collecter les résultats obtenus.[1]

- Malgré la diversité des définitions la majorité d'entre elles semblent s'accorder sur deux caractéristiques clés :

1- Les laboratoires à distance construisent des circuits à l'aide de composants réels et de mesures réelles provenant d'oscilloscopes, de multimètres et d'autres instruments, offrant aux étudiants la même expérience qu'un laboratoire en face-à-face.

2- les expériences peuvent être menées à tout moment, n'importe où, ce qui donne aux étudiants plus de flexibilité.

- L'ensemble de ces deux derniers dont apparaissent si importantes pour les laboratoires distants ont détourné l'attention de la vision traditionnelle dont les laboratoires sont menés .[1]

### 1.3 Historique des laboratoires distants :

Les laboratoires distants propagent depuis la progression rapide du réseau Internet entre le milieu et la fin des années 90s. Due à la pression financière croissante aux niveaux des universités pour fournir les expériences pratiques, les laboratoires à distance ont connu un grand avancement.[1]

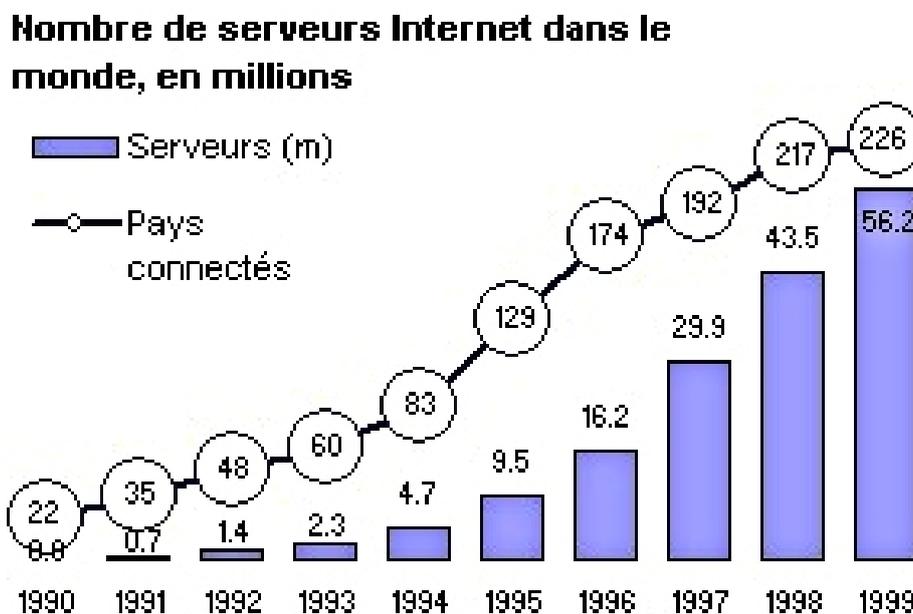


Figure1 : Evaluation de l'internet depuis 90s[20]

### 1.4 L'intérêt des laboratoires distants :

Les laboratoires à distance cherchent à résoudre de multiples problèmes rencontrés avec l'expérience de laboratoire universitaire moderne vu qu'ils :

- Augmentent la qualité d'apprentissage des étudiants en utilisant leurs temps plus efficacement.
- Offrent aux étudiants la possibilité de mener des expériences à des heures qui leur conviennent.
- Réduisent les coûts budgétaires, car les coûts de démarrage des labos distants sont moins chers par rapport aux labos classiques.
- S'adressent à un grand nombre d'étudiants.
- Donnent aux étudiants externes l'opportunité de faire les expériences en laboratoire.
- Avantage de démontrer l'équipements pédagogique localement et à l'extérieur.[1]

### 1.5 Composition des laboratoires distants :

Comme l'on a vu précédemment les laboratoires distants combinent trois

composants majeurs :

### 1.5.1 Matériel :

Présente l'environnement physique où les expériences seront exécutées par les utilisateurs distants. Il englobe tous sortes d'équipements et matériaux utilisés pendant la réalisation des expériences tel que les instruments de mesures, générateurs de signaux, des oscilloscopes et d'autres équipements spécialisés nécessaires pour effectuer les expériences. [1]

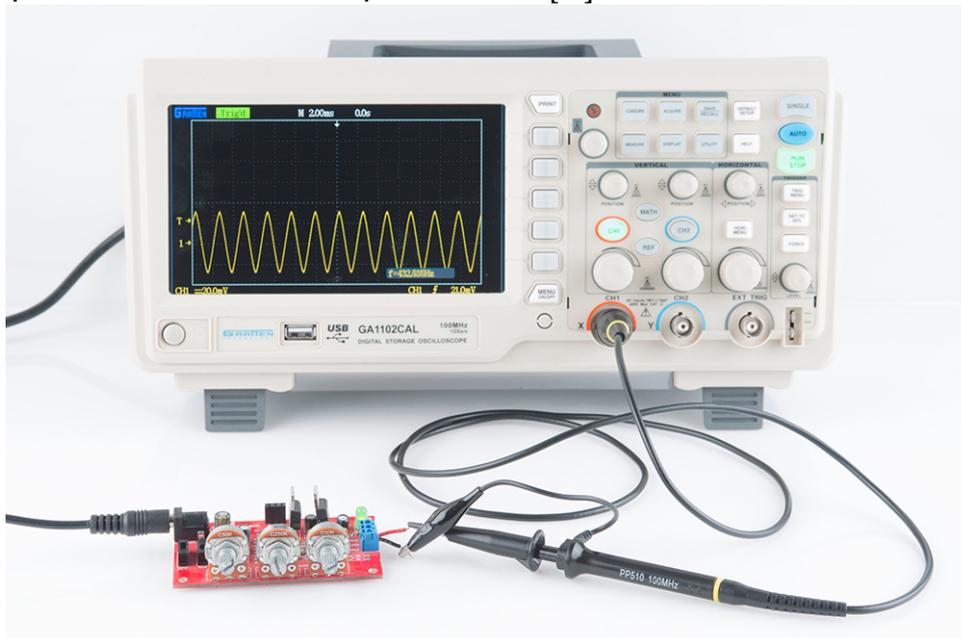


Figure2 : Oscilloscope[21]

### 1.5.2 Logiciel :

Connu par RLMS\* qui veut dire "système de gestion des laboratoires distants" c'est la partie logique du laboratoire incluant tout composant logique utilisés lors de la réalisation du laboratoire, comme l'application qui permet aux utilisateurs de y'accéder au laboratoire (notamment des applications web), les systèmes de gestion des utilisateurs, d'authentification, des expériences....[1]

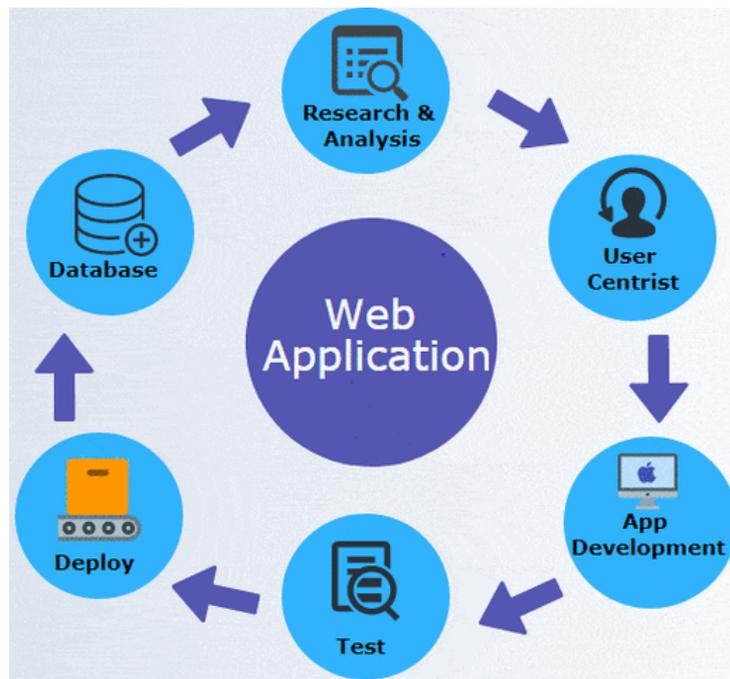


Figure3 : Cycle de vie de l'application web[26]

### 1.5.3 Communications :

La communication est nécessaire pour connecter les utilisateurs distants aux laboratoires distants. Cela peut se faire via Internet (IOS)\* "Internet of things" , ou un autre réseau de communication. On a aussi les circuits électroniques entre les composants réels, y'en a aussi les technologies telles que la vidéo-conférence pour permettre l'interaction entre les étudiants et leurs enseignants.[1]

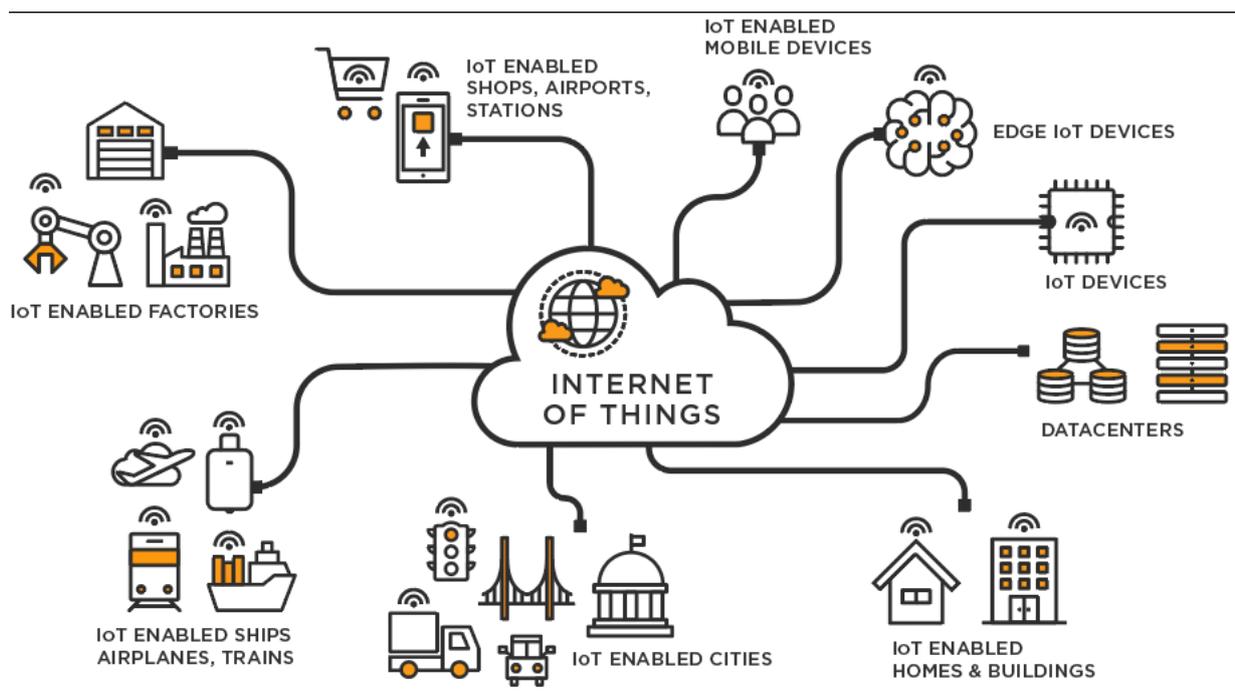


Figure4 : fonctionnalités de l'IOT[22]

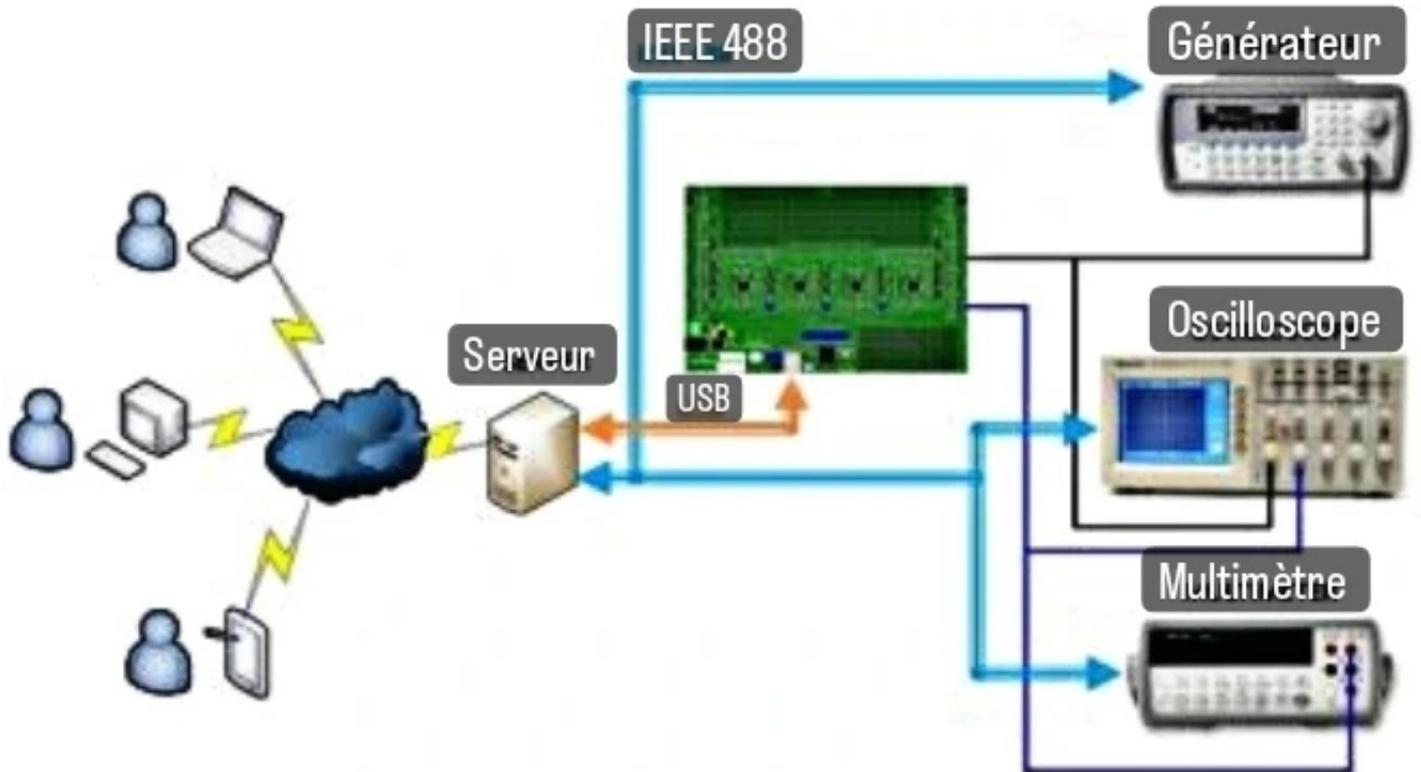


Figure5 : La composition d'un laboratoire distant[1]

## 1.6 Modes de travaux de laboratoire :

Il existe deux modes de travaux de laboratoire :

### 1.6.1 Laboratoire pratique conventionnel :

C'est le laboratoire traditionnel qu'on connaît, tout se passe dans le laboratoire avec un matériel réel et un encadrement présentiel.[3]



Figure6 : Laboratoire traditionnel[24]

## 1.6.2 Laboratoire non-conventionnel :

Il comprend deux laboratoires : le laboratoire à distance et le laboratoire virtuel.

- Le laboratoire à distance où l'étudiant de chez lui par exemple va se connecter par internet à un équipement qui se trouve dans le laboratoire et exécute les travaux demandés.
- Le laboratoire virtuel est basé sur l'utilisation du logiciel, des applications qui simulent le comportement d'un système ou d'un processus et sur lequel l'étudiant peut conduire les travaux demandés.[3]

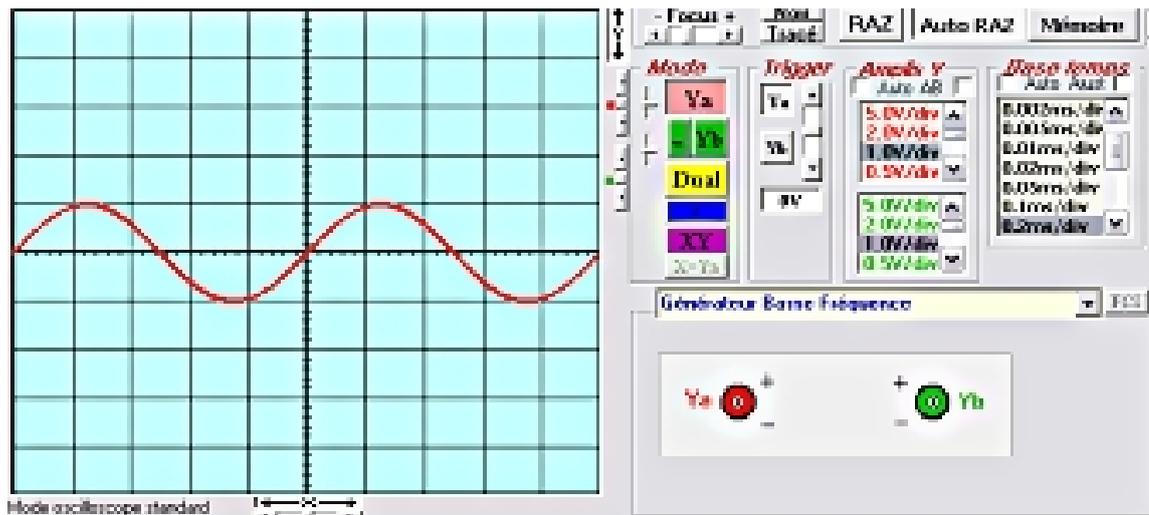


Figure7 : exemple d'un laboratoire virtuel[25]

## 1.6.3 Avantages des modes de travaux de laboratoire :

### 1.6.4 LAB pratique "traditionnel" :

- Crédibilité et confiance dans les résultats, contrôle de la réalité.
- Manipulation directe de l'équipement.

### 1.6.5 LAB à distance :

- Crédibilité et confiance dans les résultats, contrôle de la réalité.
- Coût réduit.
- Planification flexible.

### 1.6.6 LAB virtuel :

- Coût réduit.
- Faible risque.
- Planification flexible.[3]

LAB :laboratoire

Source :[https://www.abbreviationfinder.org/fr/acronyms/lab\\_laboratory.html](https://www.abbreviationfinder.org/fr/acronyms/lab_laboratory.html)

	Équipement réel	Perception étudiant(e)	Coût	Sécurité	Planification
LAB* pratique	++++	+++	+	+	+
LAB à distance	+++	+++	+++	++	+++
LAB virtuel	-	+	++++	++++	+++

**Tableau01** : "Comparaison entre laboratoires pratiques, distants et virtuels"[3]

Remarque :

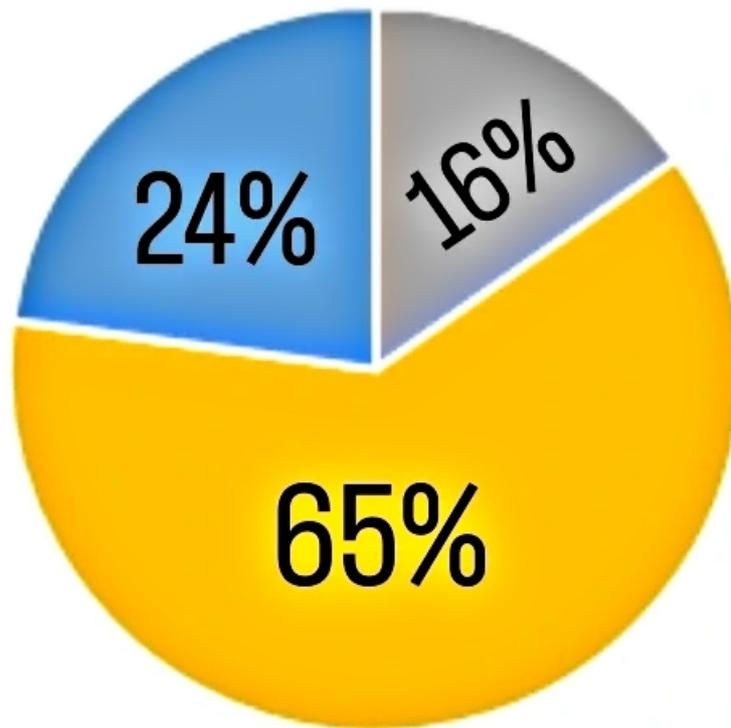
Une "Hybridation" c'est la combinaison des trois modes .[3]

## 1.7 Les apprentissages réalisés par les travaux de laboratoire :

Une méta-analyse a classé les apprentissages réalisés par les travaux de laboratoire en six points :

- 1- Connaissance et compréhension.
- 2- Habileté de recherche.
- 3- Communication sociale et scientifique.
- 4- Habileté pratique.
- 5- Habileté d'analyse.
- 6- Perception et engagement.[3]

- La plupart des études analysées montrent que les apprentissages réalisés par les travaux de laboratoires non conventionnels (virtuels ou à distance) sont égaux ou supérieurs aux apprentissages obtenus par les travaux de laboratoire pratiques conventionnels.



- Travaux pratiques non conventionnels élevés
- égalité des travaux pratiques conventionnels et non conventionnels
- Travaux pratiques conventionnels élevés

**Figure8** : étude faite sur les travaux conventionnels et non conventionnels[3]

- Pour l'analyse des apprentissages, la composante "Connaissance et Compréhension" est la plus considérée dans les études analysées.

- La perception et l'intérêt que manifeste l'étudiant(e) pour la science, vient en seconde position.

- Seulement quelques études pointent la limite du développement des habiletés pratiques dans le cas des laboratoires virtuels ou à distances.[3]

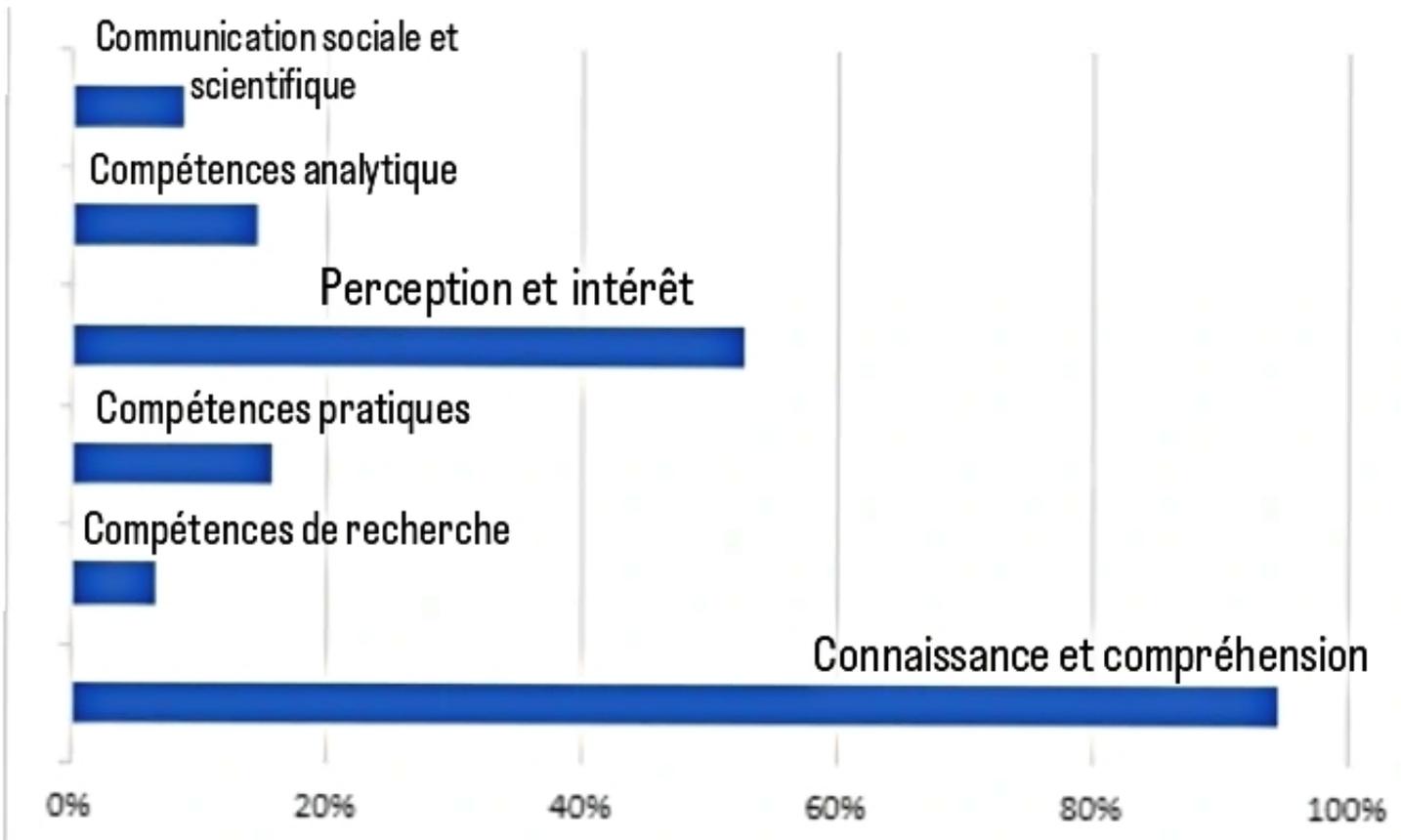
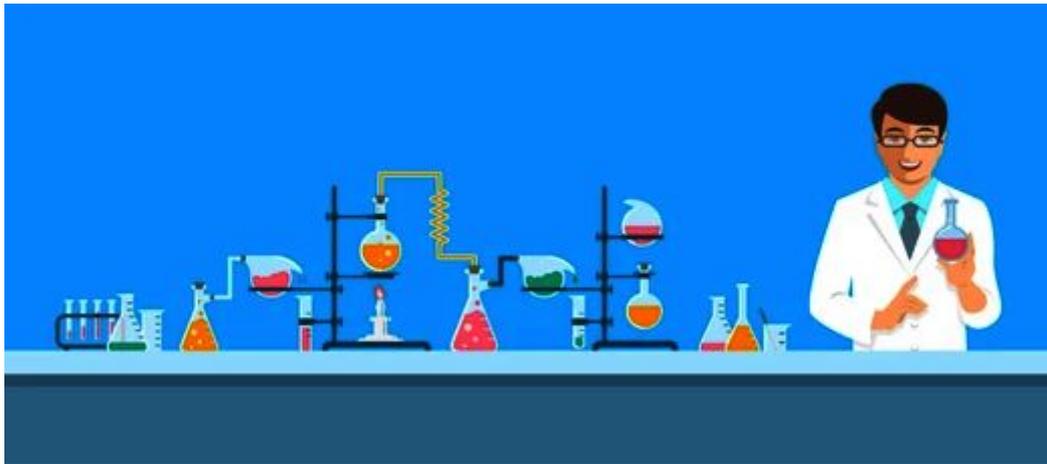


Figure9 : ensemble d'analyse des critères d'apprentissages[3]

## 1.8 Les types de laboratoires :

- Il existe deux types de laboratoires : laboratoire en technologie et laboratoire en science.
- Les laboratoires en science comprennent la physique, la chimie...Par contre les laboratoires en technologies comprennent les génies : électrique, informatique...  
Quelles sont les différences? Et quelles sont les particularités?
- Au niveau des connaissances et compréhension, pour les deux types de laboratoires, les outils de simulation virtuelle et à distance permettent de répondre aux tels objectifs.
- Pour le laboratoire en science, l'objectif de base est la "Vérification et renforcement de la théorie" tandis que pour le laboratoire en technologie y'en a des "procédures et techniques de manipulation" qui s'ajoute au niveau du laboratoire.
- Le laboratoire en science pose la question : "Es ce que l'objectif précédent peut être réaliser par les laboratoires non conventionnels?"
- De même pour les laboratoires en technologies en ajoutant cette question : "Es ce qu'on nécessite un complément pratique?"
- La créativité et le jugement des enseignant(e)s seront à l'épreuve selon la spécialité de l'enseignement concerné.[3]



**Figure10** : image représentant un laboratoire en science[27]

## 1.9 Les défis des laboratoires distants :

- Il s'avère dans la section précédente que les laboratoires distants présentent de multiples avantages, mais, ils ne forment malheureusement pas une solution simple à implémenter et ne nécessitent pas mal de ressources humaines et de temps.

- De même, la disponibilité des laboratoires à distance constitue un grand problème dans la formation des ingénieurs et cela pour deux raisons :

1- La gestion des laboratoires nécessite un entretien continu du matériel, ce qui vaut dire qu'elle peut être coûteuse en ressources.

2- La duplication (facteurs économiques) des expériences scientifiques exercés.

- Aussi un autre défi : la sécurité.

- Les laboratoires distants sont connectés au réseau mondiale "Internet", et donc ils sont vulnérables aux attaques malveillantes. Comme solution, les serveurs du laboratoire doivent être à jour, ainsi que le pare-feu, sans oublier le serveur d'expérimentation à distance dont il doit être configuré pour bloquer et signaler les activités suspectes d'une manière rapide.[1]

## 1.10 Système de gestion des laboratoires distants :

- Tout laboratoire distant a besoin d'un système pour assurer la gestion des expériences, autorisations, authentification, des files d'attente pour les utilisateurs souhaitant accéder au laboratoire, etc.

- Dans cette section nous allons présenter le concept de (RLMS) "Remote Laboratories Management System", l'un des principaux composants de l'architecture logicielle des laboratoires distants.[1]

## 1.11 Les fonctionnalités du RLMS :

Le RLMS offre une divers de fonctionnalités, dans ce qui suit nous allons voir les principales fonctionnalités de cette fondation.

- Gestion des utilisateurs : le RLMS gère les utilisateurs qui veulent manipuler les expériences, incluant leurs inscription et authentification.

- Gestion des accès et autorisations : cette dernière limite les actions effectuées par un utilisateur sur un ensemble de moyens.

- Gestion des laboratoires et d'expériences : le RLMS offre la possibilité de gérer plusieurs laboratoires géo-distribués. De plus, il offre la possibilité de la gestion de différentes expériences dans les laboratoires.

- Gestion des sessions : le RLMS gère le grand accès des utilisateurs aux laboratoires de nombres limités en même temps.

- Autre : un RLMS peut aussi inclure d'autres fonctionnalités complémentaires tel que : l'analyse de l'utilisation des laboratoires, outil d'intégration avec d'autres plate-formes tel que "Moodle"\* ..... [1]

## 1.12 Implémentation des RLMS :

Il suffit juste d'avoir un navigateur web déjà présent dans tous sorte de système d'exploitation (Windows, linux, Android...) et un appareil de type (téléphone portable, tablette, micro-ordinateur...) pour pouvoir accéder aux laboratoires distants, car ces derniers sont généralement implémentés comme des portails web ce qui offre plus de flexibilité aux utilisateurs.[1]

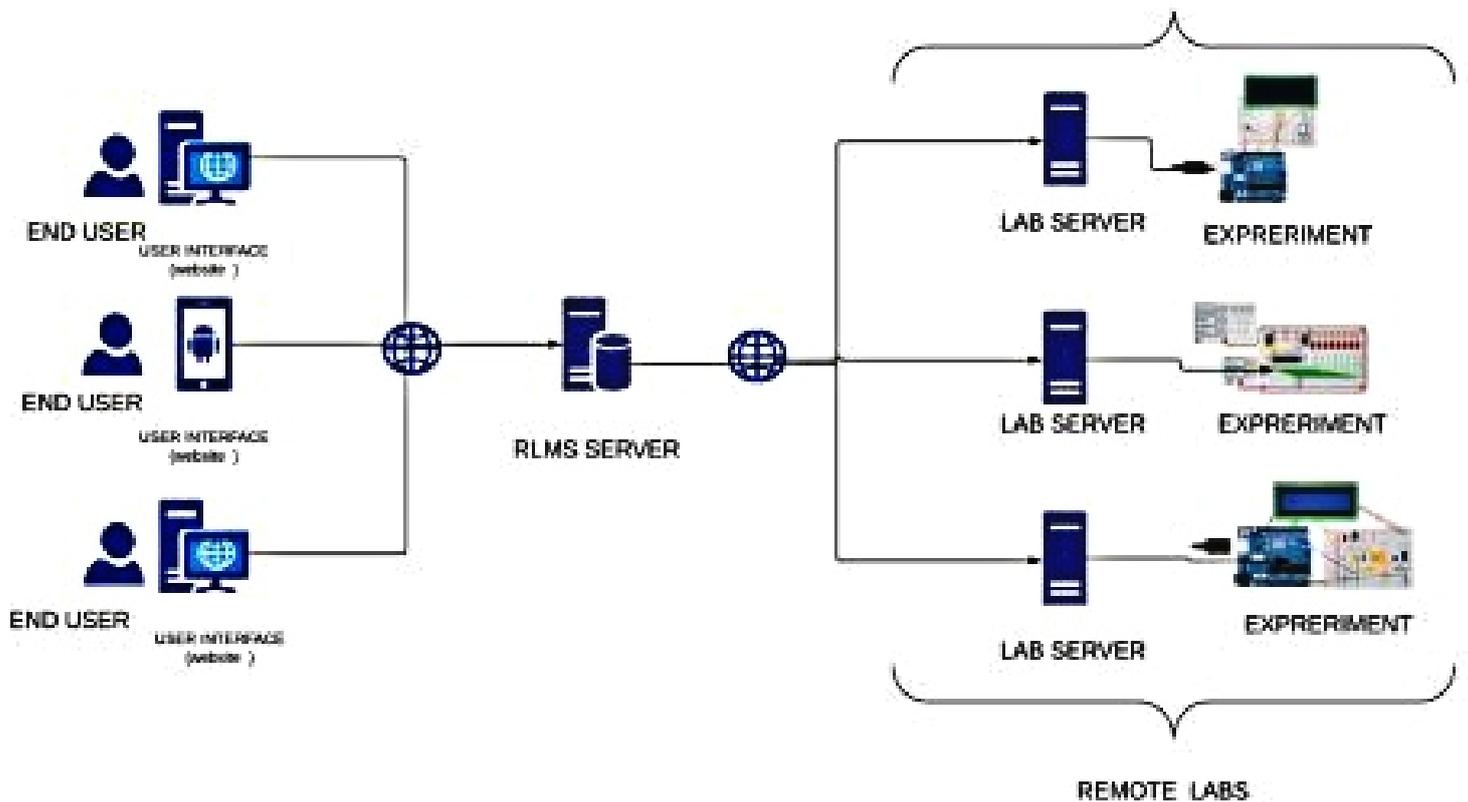


Figure11 : implémentation RLMS[1]

### 1.13 Partage des laboratoires distants :

Il existe trois approches renforçant la notion avantageuse des laboratoires distants “le partage” :

1- Laboratoire distant public : cette approche consiste à laisser le laboratoire accessible à quiconque, ce qui conduit à un compromis entre accessibilité et fonctionnalités avancés.

2- Partage des comptes : entre différentes RLMS en utilisant une authentification fédérée.

3- Laboratoires fédérés : “établissement à établissement” l’utilisation transparente d’autres laboratoire.[1]

### 1.14 Conclusion :

Notre objectif en commençant ce premier chapitre était d’explorer les domaines conceptuels et les fondements théoriques des laboratoires à distance, afin de mieux comprendre le contexte de notre recherche.

## Chapitre 2

# Évaluation des performances

## **2.1 Introduction :**

Dans le chapitre précédent, les points importants concernant la notion de "laboratoires distants" ont été abordés.

Le chapitre actuel sera une continuation, mais avec des exemples concrets afin de mieux décrire les laboratoires distants, plus spécifiquement dans le domaine de l'électronique.

## **2.2 Les étapes d'un LAD (laboratoire à distance) d'un étudiant :**

### **2.2.1 Pré-lad :**

Le processus "Pré-Lad" a pour but de donner une signification et une justification au travail qui sera réalisé en s'appuyant sur un problème authentique. Pour y parvenir, il propose une visite virtuelle qui permet de contextualiser le travail en explorant une application industrielle en ligne, ainsi qu'une exploration du matériel pour renforcer les connaissances technologiques grâce à des illustrations telles que des vidéos et des images.

### **2.2.2 Lad :**

Le processus "Lad" comprend des tests élémentaires, la réalisation des travaux avec des consignes de sécurité rigoureuses étape par étape pour le travail à distance.

### **2.2.3 Post-lad :**

Après le processus "Lad", le rapport permet de vérifier les nouvelles compétences acquises grâce au processus "Lad" et d'ouvrir de nouvelles perspectives sur des problèmes à résoudre. Des critères d'évaluation sont également définis pour mesurer le succès du processus.

### **2.2.4 LAD en cas d'impossibilité d'accès aux équipements par l'étudiant :**

Ici on envisage des options :

Option 1 : Filmer et enregistrer la manipulation et fournir la vidéo et les mesures effectuées.

Option 2 : Prendre des photos des étapes importantes de la manipulation et fournir les mesures effectuées.

Option 3 : Diffuser en direct la réalisation de la manipulation et partager les mesures effectués.

Option 4 : Proposer différentes versions et résultats de réalisation du travail de

lab, et demander à l'étudiant d'identifier les différences et les erreurs.[4]

## 2.3 Quelques travaux de LAD des pays du monde :

### 2.3.1 Lab@home :

- "Laboratoire chez soi". Il est basé sur les équipements miniatures que les technologies permis aussi et qui font les fonctionnalités de plusieurs instruments en même temps en utilisant un logiciel approprié qui remplace différents instruments et exerce leurs fonctions.[3]

- A partir d'un petit kit on peut remplacer des générateurs, des appareils de mesures et obtenir et conduire les travaux de laboratoire classique chez-soi.[3]



Figure12 : image représentant un logiciel qui remplace différents instruments électroniques[3]

### 2.3.2 L'innovation à l'UQO :(Université Québec en Outaouais)

L'expérience de l'UQO\* dans sa deuxième version, est inspirée du principe du lab@home qui utilise des kits miniatures mis à la disposition de l'étudiant chez lui pour réaliser les travaux pratiques (TP).

- La photo dessous montre deux instruments :

WaveForms : logiciel assurant des fonctions équivalentes de 12 instruments(générateur,

LAD : laboratoire à distance Source :[https://ceea.ca/wp-content/uploads/2021/02/GREC\\_Laboratoires\\_a\\_distance.pdf](https://ceea.ca/wp-content/uploads/2021/02/GREC_Laboratoires_a_distance.pdf)

oscilloscope, alimentation..)

KIT : Digilent Analog Discovery Studio.[4]

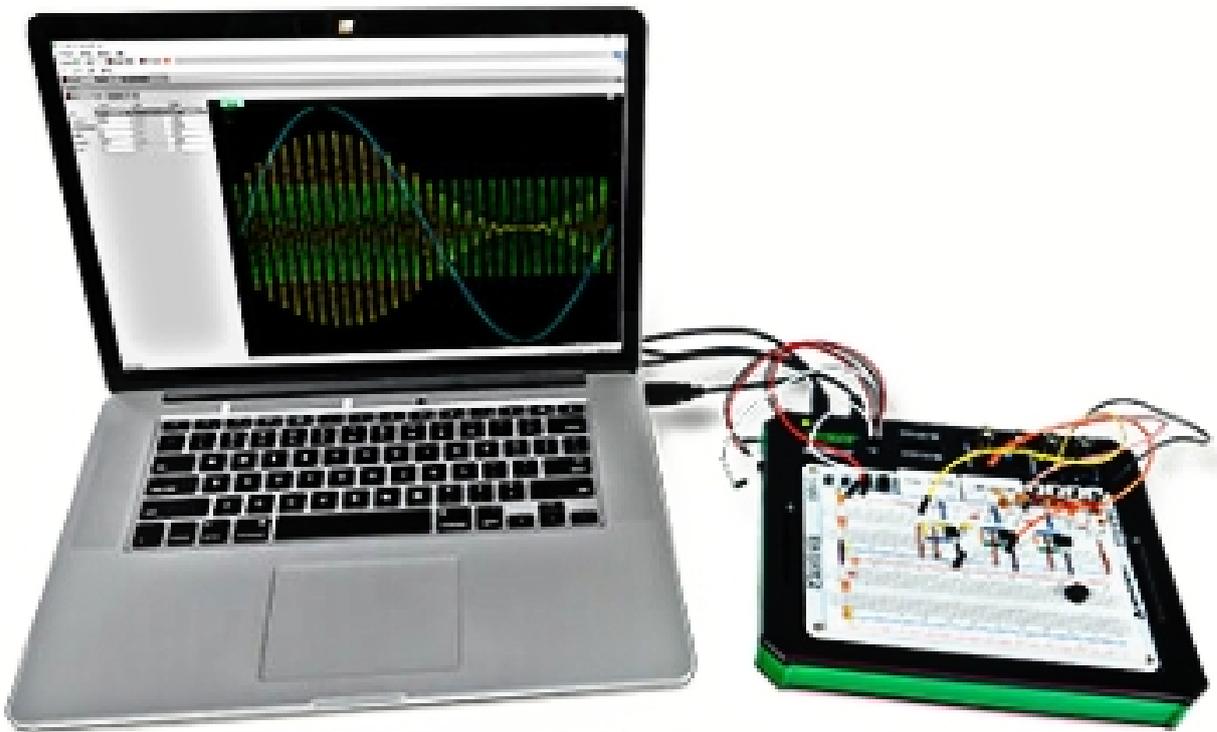


Figure13 :réalisation d'un laboratoire distant[4]

Voici une idée sur le dispositif de manipulation, avec le Kit. On peut disposer d'une caméra supplémentaire pour visualiser ce qui se passe au niveau du montage du kit.D'autre équipements complémentaires peuvent être ajoutés selon la nécessité du travail( composants, appareils de mesure spécifiques...).[4]



**Figure14** : dispositif de manipulation[4]

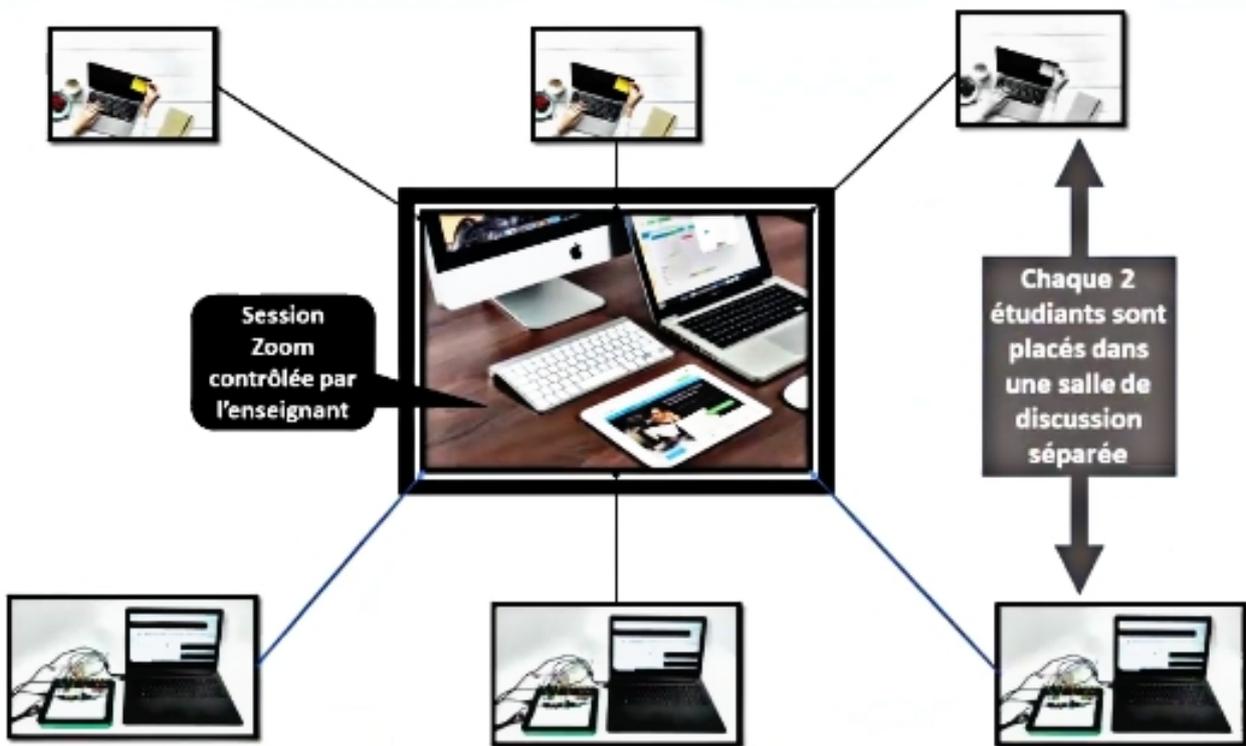
Quelles sont les étapes pour la conduite de cette deuxième version du LAD de l'UQO?

1-Les étudiant(e)s sont réparti(e)s en équipes de deux. Les membres de chaque équipe sont à des endroits différents (un au laboratoire et l'autre chez lui).

2-Un(e) des membres de l'équipe est un possesseur d'un Kit, l'autre n'en a pas .

3-L'enseignant(e) démarre une session Zoom à laquelle tous les étudiant(e)s sont invité(e)s.

4-L'enseignant(e) place chaque équipe dans une salle (breakout rooms de Zoom) distincte.[4]



**Figure15** : LAD de l'UQO[4]

L'enseignant(e) ouvre une session de Zoom pour accueillir tous les étudiant(e)s et placer chaque équipe dans une salle séparée.

- L'étudiant(e)(A) disposant du kit réalise le montage du circuit.
- L'étudiant(e)(B) suit ces opérations à travers l'image transmise par la caméra. Il peut aussi interagir et discuter avec son co-équipier(A).
- L'étudiant(e)(A) lance le logiciel "WaveForms" et partage son écran avec son co-équipier(B).
- L'étudiant(A) commence la réalisation de la première partie des travaux en utilisant le logiciel du Kit WaveForms. Le deuxième étudiant(e)(B) vérifie, prend les mesures, discute avec son co-équipier et note les commentaires.[4]

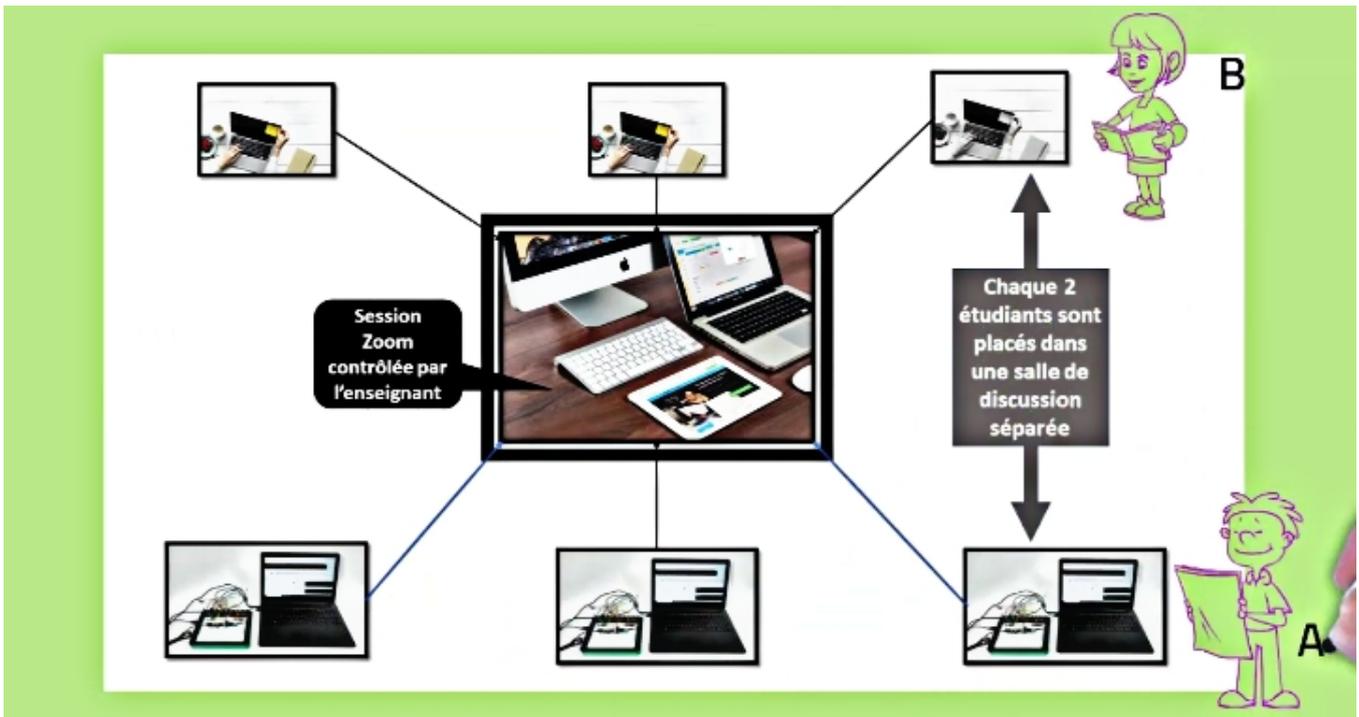


Figure16 :La session Zoom réalisé par l'enseignant[4]

"Permutation des rôles pour la partie suivante de la manipulation :"  
 Une fois la première partie terminée, l'étudiant(e)B demande la prise de contrôle de l'écran de(A).  
 Les rôles seront permutés et l'étudiant(e)(B) pourra alors réaliser la deuxième partie en contrôlant à partir de son ordinateur la logiciel "WaveForms".  
 La permutation des rôles se poursuivra pour les parties suivantes de la même façon.[4]

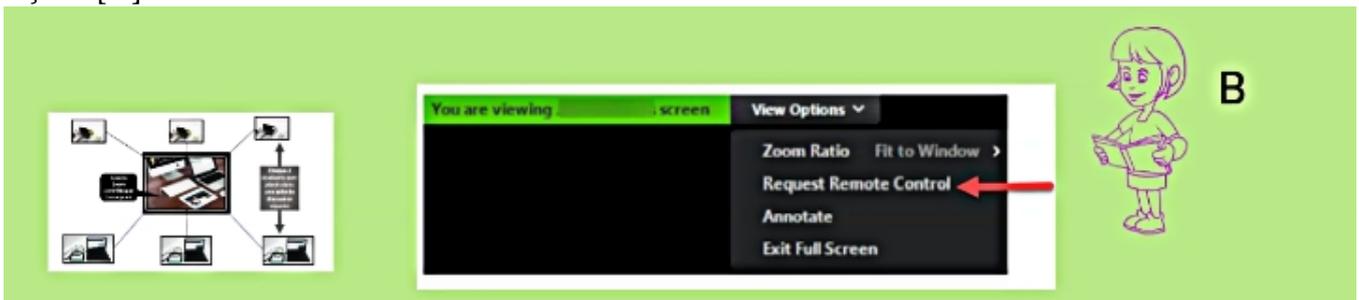


Figure17 : deuxième partie de la manipulation[4]

L'enseignant supervise le déroulement de tout le processus à partir de sa session Zoom et peut passer d'une salle (breakout room) à une autre et vérifie le bon déroulement du travail.  
 Les étudiant(e)s peuvent aussi inviter l'enseignant(e) à les rejoindre dans leurs salles pour répondre à des questions et apporter du soutien.[4]



Figure18 : Soutien de l'enseignant en rejoignant les salles des étudiants[4]

A la fin des travaux sur le kit, les étudiant(e)s pourront continuer à discuter les résultats et les détails du rapport à remettre.[4]

- Perspectives 1/2

La réussite de cette expérimentation de laboratoire à distance(LAD) à l'échelle d'un groupe d'étudiants, prouve la possibilité de réaliser des séances de travaux de laboratoire totalement à distance.

Ceci représentait une solution de secours dans la situation de la pandémie, mais continue encore à représenter une démarche précieuse dans le cas de formation à distance dans les spécialités scientifiques et technologiques.[4]

- Perspectives 2/2

L'impact de cette démarche innovante peut être très important si on considère ce qu'elle apporte comme libération des espaces de laboratoire et réduction sur le coût des équipements qu'on aurait mobilisé.

Il est toutefois important de tracer une stratégie précise pour prévoir des opportunités complémentaires pour que l'étudiant(e) découvre aussi les instruments classiques et maîtrise leurs utilisations.

La prochaine étape serait de planifier l'utilisation du LAD à une échelle plus large et de revoir l'articulation du LAD avec les outils exercices de simulation et le cours

théorique.[4]

### 2.3.3 Projet laboratoire à distance LAD (Remote lab) à Montréal :

- Le projet a été conduit par 3 Cégep : Valleyfield, Sorel-Tracy, Granby, piloté par l'école supérieur de technologies qui ont développé 24 expériences de laboratoires à distance.

- Le Cégep A pouvait utiliser l'équipement du Cégep B et vice-versa.
- L'intéressant dans ce travail c'est que l'étudiant pourrait à partir de chez lui se connecter aux matériels et réaliser le travail demandé. Il va avoir à travers une vidéo une visualisation de ce qui se passe sur l'équipement qu'il contrôle, puis la collecte des résultats et des mesures sur lesquelles il peut faire son rapport.[3]

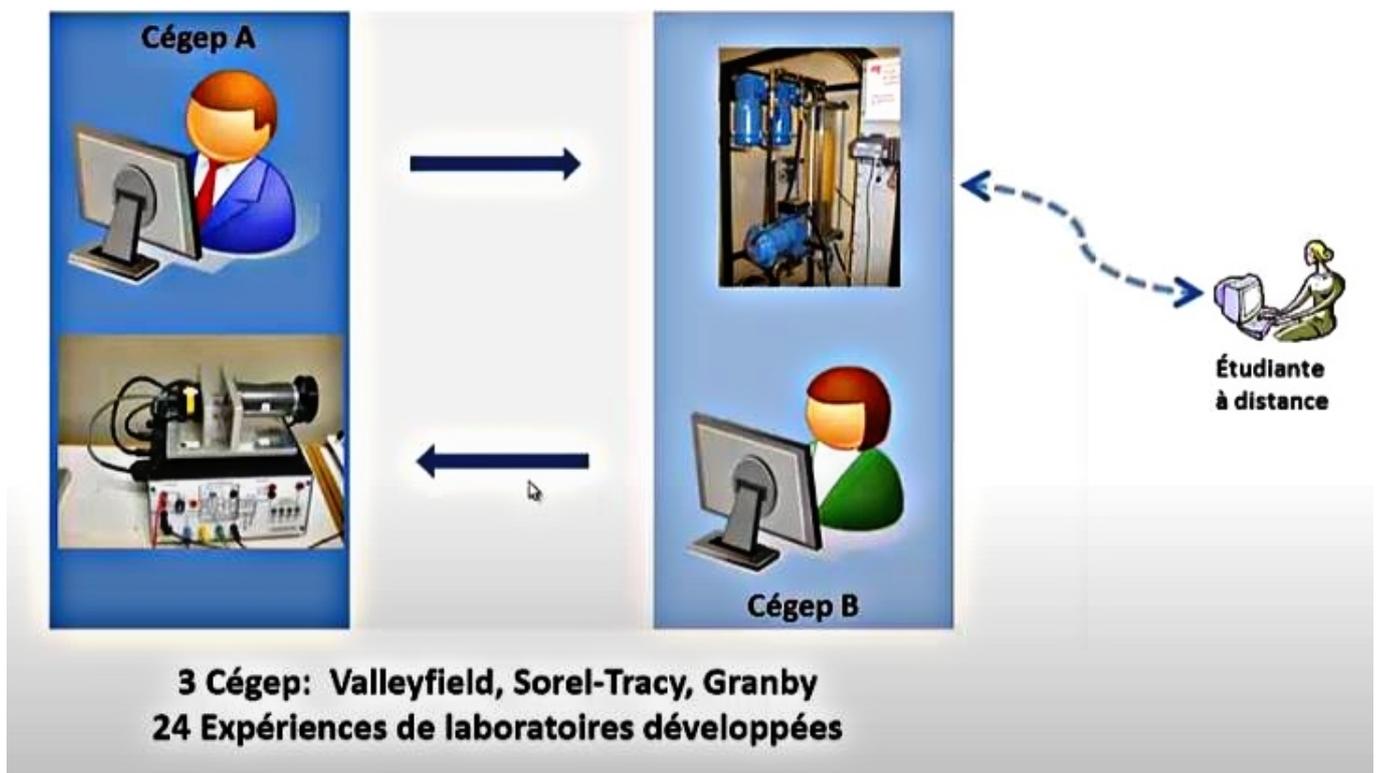


Figure19 : Remote lab à Montréal[4]

### 2.3.4 Ilab Shared Architecture :

L'architecture ILab se concentre sur le développement rapide de laboratoires indépendants de la plate-forme, une gestion efficace pour les fournisseurs de laboratoires tout en préservant l'autonomie des enseignants et un accès évolutif pour les étudiants.

Il s'agit d'une boîte à outils logicielle distribuée et une infrastructure de services middleware pour prendre en charge les laboratoires distants et promouvoir leur partage entre les écoles et universités du monde entier.

Son architecture ressemble beaucoup à l'architecture web à trois niveaux :

- Le premier niveau est l'application cliente de l'étudiant, elle s'exécute généralement sous forme d'applet ou d'application téléchargées sur le poste de travail de l'étudiant.
- Le deuxième niveau "intermédiaire", appelé service "courtier" , fournit les services communs partagés. Il s'appuie sur une base de données relationnelle standard telle que Microsoft SQL Server ou MySQL. Le client de l'étudiant communique seulement avec le service "courtier", qui transmet les spécifications de l'expérience au niveau final.
- Le niveau finale est le serveur du laboratoire, qui s'interface avec les instruments qui exécutent les expériences spécifiées. Le serveur de laboratoire avertit le service courtier lorsque les résultats sont prêts à récupérer.

Dans cette architecture , le client et le serveur représentent tous deux le modules logiciels dépendant du domaine et du laboratoire.

Le service "courtier" est un code complètement générique et peut interagir avec n'importe quelle combinaison de client et serveur de laboratoire qui implémentent les interfaces appropriées exprimées en termes d'appels de l'SOAP\* définis dans WSDL\*.[1]

---

SOAP :Simple Object Access Protocol <https://www.ibm.com/docs/en/sc-and-ds/8.1.0?topic=stack-simple-object-access-protocol>

WSDL :Web Services Description Language [https://www.ibm.com/docs/en/radfw/9.6?topic=SSRTLW\\_9.6.0/org.eclipse.jst.ws.doc.user/concepts/cwsdl.htm](https://www.ibm.com/docs/en/radfw/9.6?topic=SSRTLW_9.6.0/org.eclipse.jst.ws.doc.user/concepts/cwsdl.htm)

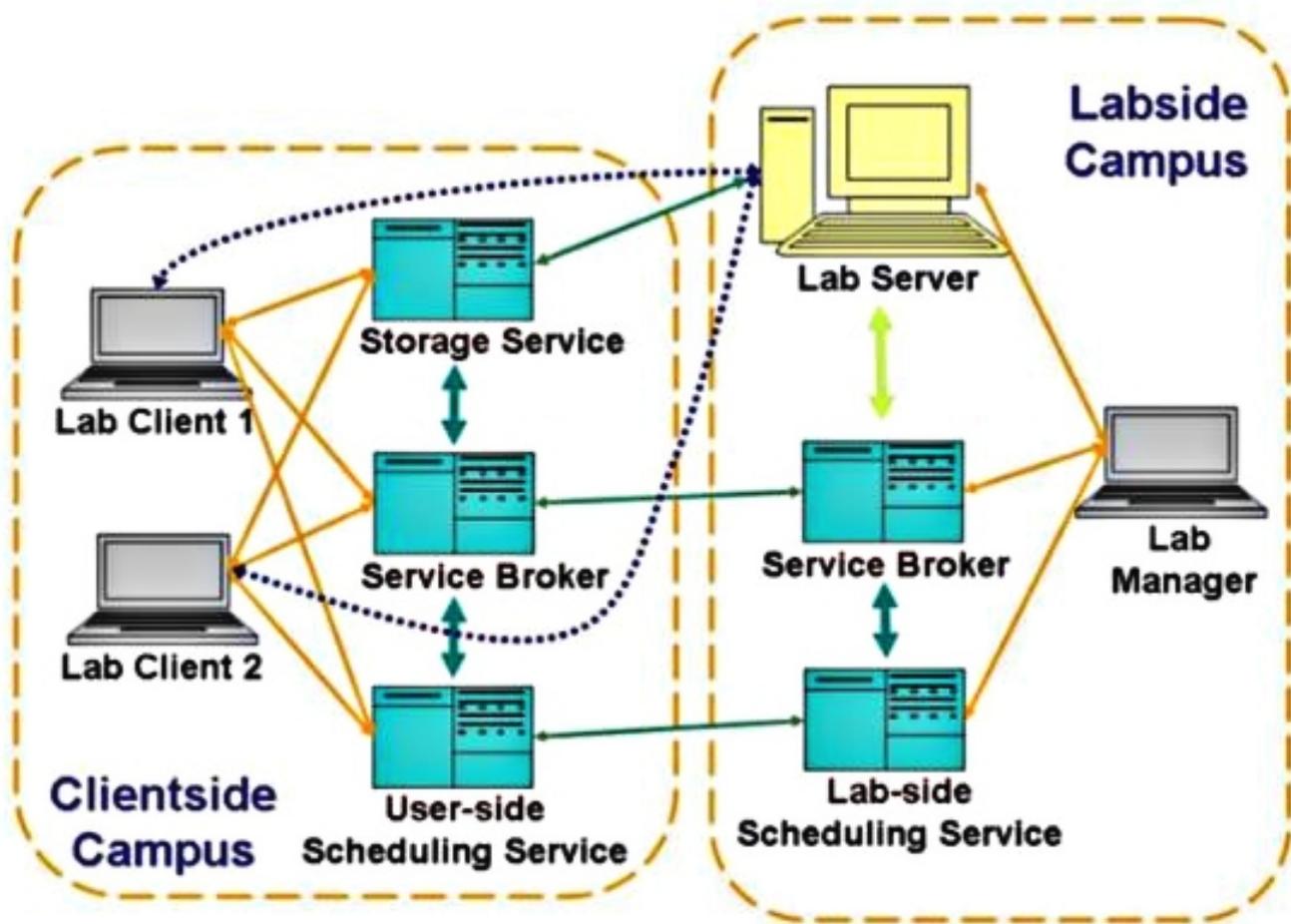


Figure20 : topologie architecturale de iLab[2]

### 2.3.5 RemLabNet :

Il est destiné à la gestion d'expériences à distance pour les nouveaux étudiants des niveaux universitaires et secondaires due d'une absence similaire en Europe, construit à l'aide de nouveaux composants ; tel que : un entrepôt de données, la carte de communication du RLMS...etc.

- Le serveur de communication fournit également des services de connexion et de diagnostic, ainsi que des services destinés au confort de l'enseignant (tableau blanc, IP) téléphonique, gestion des tests et des réservations, inclusion de la simulation.
- Pour certains raisons de sécurité, un accès optimal à toutes les expériences économique du cloud virtualisé seront utilisés.
- L'état actuel des expériences à distance dans ce projet de laboratoire électronique est accessible via la page Web du projet (<http://www.ises.infos>); Contrôle du niveau d'eau, induction électromagnétique, etc.[1]

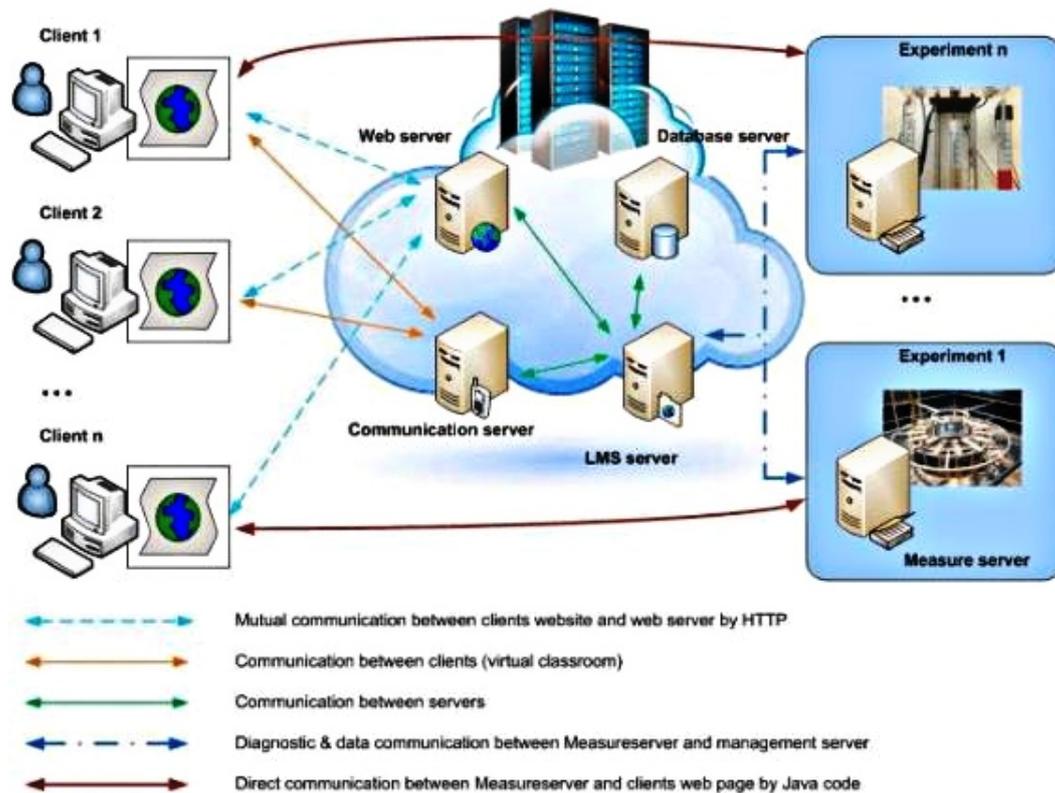


Figure21 : Fonctions générales de RemLabNet[1]

### 2.3.6 Weblab : Université de Deusto en Espagne

Le projet Weblab Deusto est un effort commun pour donner accès à un vrai laboratoire d'équipement sur Internet à des fins de recherche et d'enseignement. Ça peut être catégorisé comme étant un système de laboratoire à distance. Du point de vue cognitif, les supports d'apprentissage de Weblab Deusto sont présentés sur une interface utilisateur Web. Il est conçu de manière à ce que les matériaux ne soient pas présentés sur la même interface que l'interface distante de l'expérience. Pour cette raison, la description de l'expérience à distance et d'autres les matériels d'apprentissage doivent être rassemblés et consultés séparément par les utilisateurs.[1]

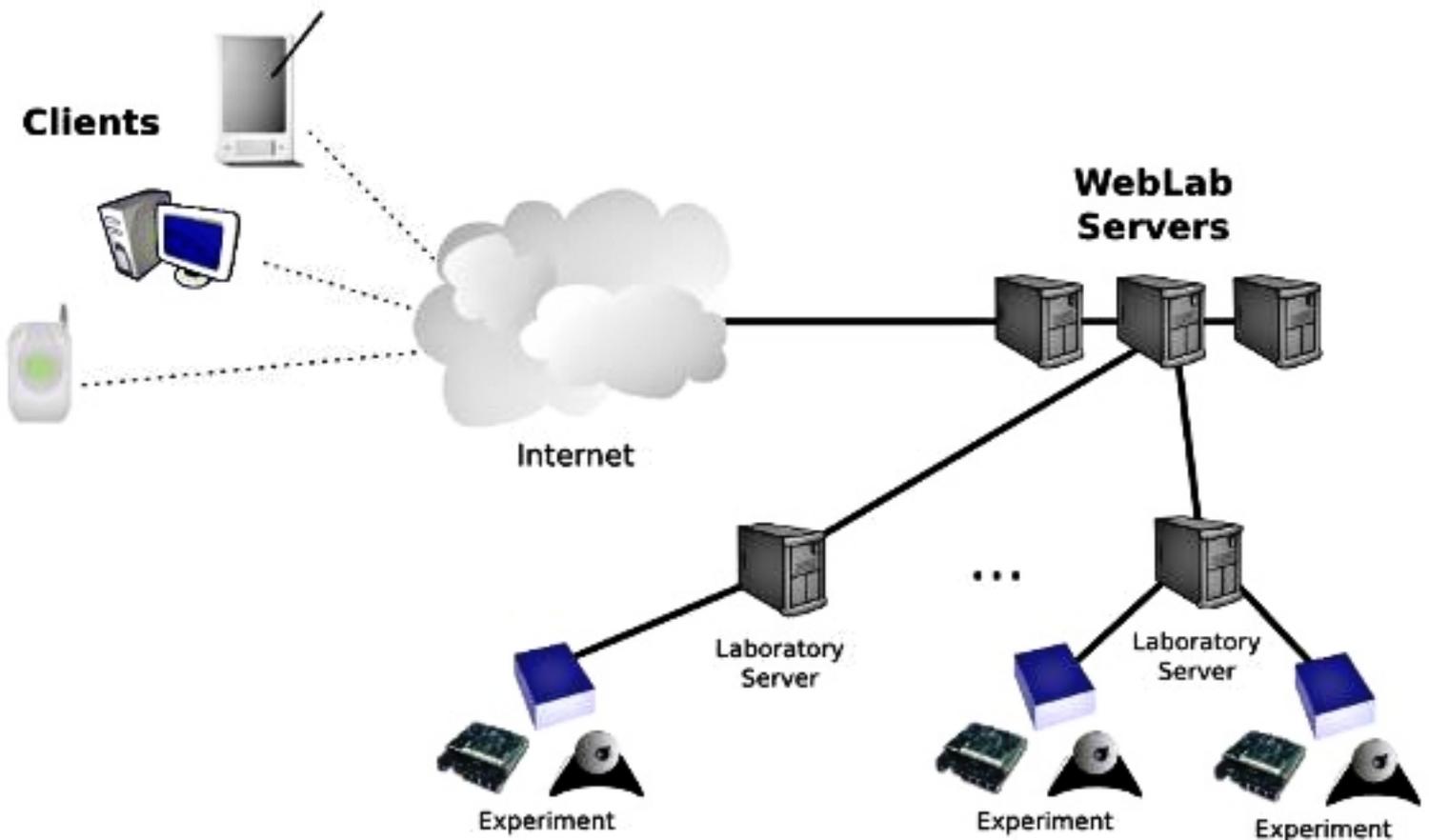


Figure22 : l'architecture du weblab[1]

### 2.3.7 VISIR (Virtual Instrument Systems in Reality) : Blekinge Institute of Technology (BTH) in Sweden 2015 :

- L'unité de traitement du signal (ASB) de l'Institut suédois de technologie (BTH) a mis en place un laboratoire à distance appelé VISIR pour les expériences électriques.
- L'interface VISIR permet aux étudiants de concevoir et de tester leurs propres circuits à l'aide des dispositifs et des composants réels tels que des oscilloscopes.
- Ce laboratoire ouvert offre aux enseignants et aux étudiants la possibilité d'effectuer à distance des expériences pratiques dans le monde réel à l'aide d'équipements de mesure et de test que l'on trouve couramment dans les laboratoires électroniques.
- Les écrans d'ordinateur sont utilisés par l'utilisateur pour faire fonctionner les panneaux de commande, les planches à pain d'instruments, et les connexions électriques qui sont effectuées via une matrice de commutateurs de relais dans laquelle les composants fournis sont installés.[6]



Figure23 : interface de Visir[6]

## 2.4 Les travaux de LAD en Algérie :

## 2.5 MostaLab -Mostaganem- :

Mostalab est une solution ductible pour but de faciliter les tâches reliées à la problématique de la gestion des laboratoires distants.

### 2.5.1 Contraintes fonctionelles de Mostalab :

Lors de la conception de la solution, certaines contraintes ont été respectées :  
*1ère contrainte* : Les fonctionnalités basiques du système : "Gestion des laboratoires", "Gestion des utilisateurs des laboratoires", "Gestion des accès aux laboratoires" sont assurées.

*2ème contrainte* : L'intégration des nouveaux laboratoires ont été assurée.

*3ème contrainte* : La montée en charge "scaLability".

Ce genre de système est un système très évolutif, donc il est conçu d'une manière optimisée.

### 2.5.2 Contraintes non-fonctionelles : "techniques"

Pendant le développement de la solution quelques contraintes ont été prise en considération :

- La solution proposée assure la disponibilité, portabilité aux utilisateurs (portail web), et la maintenabilité.
- L'interface de l'application est ergonomique.
- Prendre en considération la sécurité du système dès le début de la conception.

### 2.5.3 Besoins fonctionelles :

La solution permet la :

- Gestion des laboratoires : Ajouter, supprimer, modifier des laboratoires distants.
- Gestion des utilisateurs des laboratoires : Inscrire, modifier, supprimer des nouveaux profils des utilisateurs.
- Gestion des catégories et instances des laboratoires : Regrouper les laboratoires en catégories (dans le cas où il existe plusieurs instances du même laboratoires).
- Gestion des réservations des laboratoires : l'utilisateur accède aux instances des laboratoires.[1]



Figure24 : Exemple d'un déploiement de MostaLab[1]

## 2.6 Vue sur quelques fameux laboratoire virtuel : MSOL, LABER-SIME :

### 2.6.1 1-MSOL (Massively scalable online laboratories)

La plate-forme MSOL est une application Web qui affiche les expériences téléchargées de manière interactive, fournissant des outils de collaboration sociale.

Un étudiant intéressé à en savoir plus sur la diffraction est accueilli par une page de destination où il peut lire l'idée générale de la plate-forme MSOL et parcourir le catalogue d'examens.

Cette interface reproduit donc très fidèlement l'expérience réelle. L'utilisateur peut participer à l'expérience comme s'il était au laboratoire,

- Tout d'abord, après le lancement du laboratoire, un résumé des informations les plus importantes s'affiche et la possibilité de suivre un didacticiel qui guide l'utilisateur à travers l'interface. Ce didacticiel fait référence aux différents boutons de l'interface utilisateur et explique comment utiliser la plate-forme MSOL, et les informations (résumé, théorie et données de test) sont affichées à droite sous les images d'examen.

- Deuxièmement, et très important pour l'idée derrière la plate-forme MSOL, il existe de diverses fonctionnalités sociales disponibles accessibles via une superposition séparée et affichée avec des boutons de fonctionnalité. La principale fonctionnalité sociale est la possibilité de créer des réunions avec d'autres utilisateurs. De cette façon, les gens peuvent partager leurs expériences et explorer ensemble la physique sous-jacente à l'expérience. Incluant la capacité de partager des émotions.[7]

## 2.6.2 2- LABERSIME (École polytechnique d'Agadir-Maroc) :

- Le laboratoire d'innovation durable et recherche appliquée à l'école polytechnique d'Agadir a développé une plateforme à faible coût installée en cloud sous le nom de "LABERSIME" et dotée d'un système embarqué pour piloter les équipements réels du laboratoire, d'effectuer des expériences qualitativement efficaces que celles en mode présentiel, afin d'assurer les interactions entre les apprenants et les équipements du laboratoire (capteurs, actionneurs, instruments de mesures...).

- Le principe de la plateforme est assez simple : un apprenant qui se trouve à distance des ressources expérimentales se connecte au serveur via l'interface d'un ordinateur ou d'autre dispositif (tablette...) à l'Internet, trois entrées sont possibles.[5]

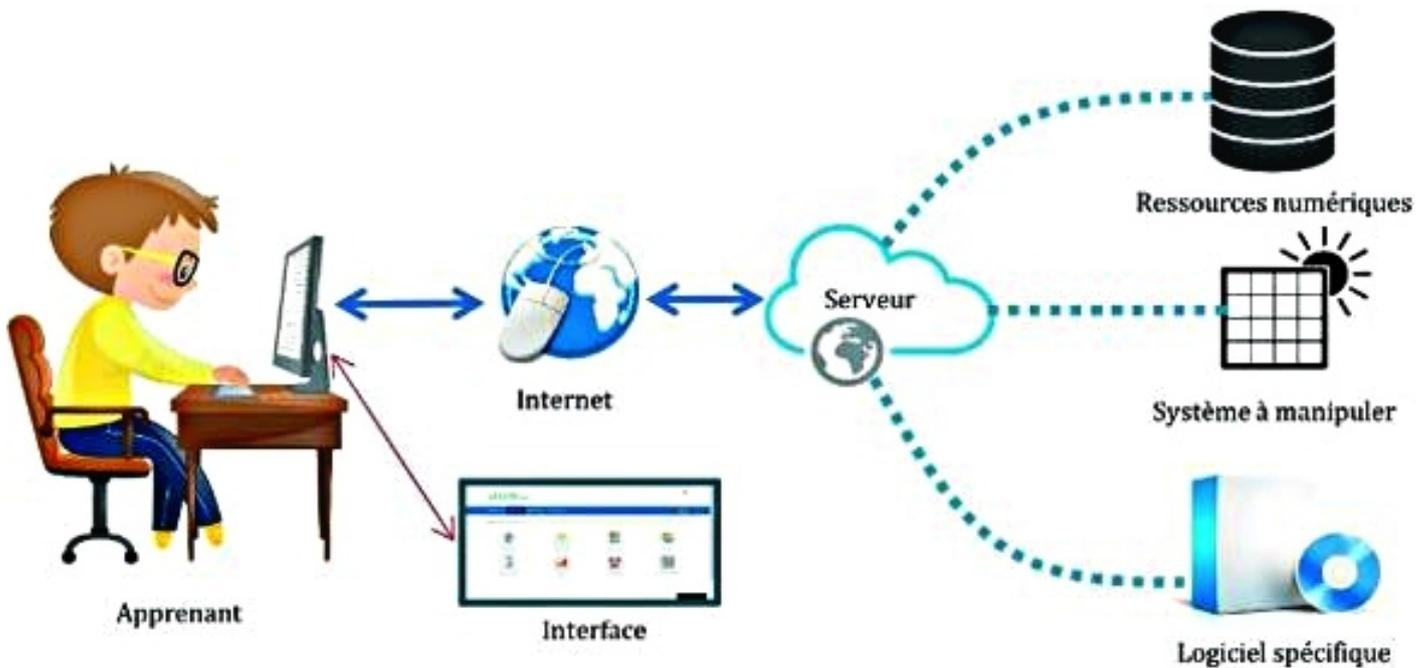


Figure25 : LABERSIME interactions[5]

## 2.7 SQL et NOSQL :

### 2.7.1 La base de données SQL :

La base de données SQL d'un laboratoire électronique est un moyen d'organiser et de stocker les données de mesure et de test des instruments, elle permet d'inclure des tableaux de machines, de tests, de mesures et de résultats. Les requêtes SQL\* peuvent être utilisées pour la récupération des données sur une machine, pour trouver chaque test effectué sur la machine ainsi la recherche des résultats de mesure pour un tel test. Cette base de données peut être utilisée pour le suivi de l'état des appareils, les tests réalisés et les résultats des tests obtenus.

- Parmi les systèmes de gestion de base de données SQL les plus connus on trouve : MySQL, PostgreSQL, SQLite, Oracle Database, MicrosoftSQLServer, Firebird, Ingres.

### 2.7.2 Base de données NOSQL :

Une base de données NoSQL dans le contexte d'un laboratoire électronique peut être une solution appropriée pour la gestion et la conservation des données non structurées de grandes quantités, telles que des mesures, des images, des signaux, etc. Les bases de données NoSQL sont souvent utilisées pour les applications qui nécessitent une évolutivité horizontale, une disponibilité élevée et une

haute performance.

Les requêtes dans une base de données NoSQL peuvent être utilisées pour récupérer des données sur une machine spécifique, pour but de repérer tous les tests effectués sur la machine ainsi la recherche des résultats de mesure pour un certain teste . Cette base de données pourrait être utilisée pour gérer et stocker des données de mesures d'une manière progressiste, flexible et sans être limitée par une structure de données prédéterminée.

- Parmi les systèmes de gestion de base de données orientés documents les plus connus on trouve : BaseX, CouchDB, Elasticsearch, eXist, MongoDB, SimpleDB.

## 2.8 Comparaison entre SQL et NOSQL :

Voici ci-dessous quelques comparaisons entre Ses deux derniers importants notions :

Structure de la base de données : le "SQL" a des structures de données stables et fixes, tandis que le "NoSQL" les données sont plus flexibles et ne possède pas de structure de données fixe.

Dans un laboratoire électronique, les données peuvent être structurées de même elles peuvent également être non structurées. Une base de données SQL est pour but de stocker des données structurées tandis qu'une base de données NoSQL est utilisée pour le stockage des données non structurées.

Évolutivité : les bases de données NoSQL sont souvent plus évolutives que les bases de données SQL car elles peuvent être facilement distribuées sur plusieurs serveurs. Ceci est important dans les laboratoires d'électronique ou de grandes quantités de données doivent être stockées ou traitées rapidement.

Performance :En terme de rapidité Les bases de données NoSQL sont souvent plus rapides que les bases de données SQL pour les opérations de "lecture/écriture" puisqu'elles n'ont pas à passer par des étapes comme le font des bases de données SQL. Dans les laboratoires d'électronique, cela peut être important lorsque les données de mesure doivent être analysées rapidement ou que les données doivent être traitées en temps réel parallèlement .

Langage de requête : les bases de données NoSQL utilisent un langage propre à chaque système de gestion de base de données (par exemple, MongoDB utilise le langage de requête BSON) contrairement au langage SQL dont il est utilisé pour les requêtes dans les bases de données SQL. Dans un laboratoire électronique, cela peut se révéler important en cas dans vous êtes déjà familier avec SQL ou vous devez travailler avec des systèmes qui utilisent les requêtes SQL.

En résumé, le choix entre les bases de données SQL et NoSQL dans un laboratoire d'électronique dépend des besoins spécifiques du laboratoire, de la nature des données, des compétences techniques et des ressources informatiques disponibles.

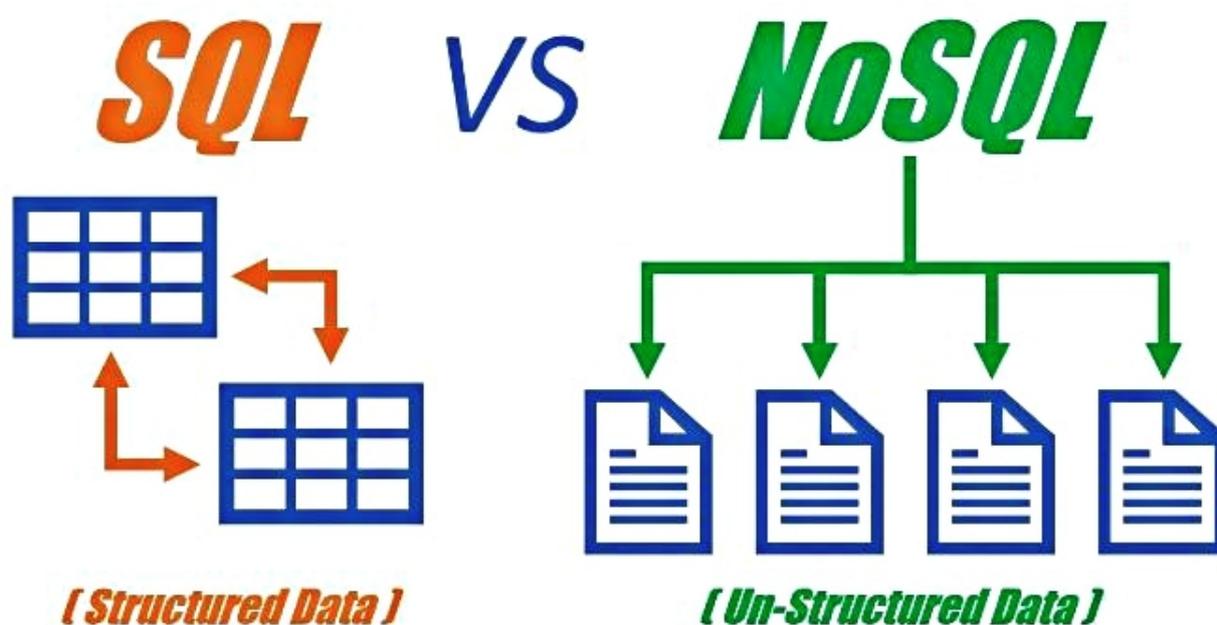


Figure26 : SQL Vs NoSQL[23]

## 2.9 Conclusion :

Pour conclure, ce chapitre a donnée une vision plus proche sur les laboratoires distants et quelques notions (SQL et non SQL), afin de permettre de se diriger vers le dernier chapitre le plus crucial d'implémentation.

## **Chapitre 3**

### **Étude comparatif de la performance de R-LAB**

## 3.1 Introduction :

- Le dernier chapitre de ce travail représente la conclusion finale de notre travail, cette section présente la mise en œuvre concrète de l'étude, y compris les outils utilisés, et enfin procède à une analyse comparative détaillée.

## 3.2 Les outils utilisés :

### 3.2.1 Visual Studio Code :

Visual Studio Code est un éditeur de code simplifié et extensible qui est gratuit développé par Microsoft pour Windows, Linux et macOS.

### 3.2.2 Fonctionnalités de Visual Studio Code :

- La prise en charge du débogage.
- La mise en évidence de la syntaxe.
- La complétion intelligente du code, les snippets et la refactorisation du code et Git intégré.[8]

### 3.2.3 Overleaf :

Overleaf est un éditeur LaTeX en ligne, collaboratif en temps réel.[9]

### 3.2.4 Latex :

LaTeX est un langage et un système de composition de documents.

### 3.2.5 Fonctionnalités du Latex :

- Réaliser des documents dont la mise en page est faite automatiquement en se conformant à des normes typographiques.[10]

### 3.2.6 Artillery.io :

Artillery.io est une solution avancée de test de charge conçue spécialement pour les équipes inter-fonctionnelles qui travaillent sur des systèmes basés sur les microservices. Son objectif principal est d'évaluer et de mesurer divers aspects de performance des systèmes. Les développeurs utilisent Artillery.io pour modéliser le comportement des entités ou des utilisateurs qui interagissent avec un système donné, afin de vérifier que celui-ci se comporte conformément aux attentes en termes de fonctionnalités et de performances, même lorsque soumis à des

contraintes de charge.

Le moteur d'Artillery.io est capable de générer une charge en utilisant différents protocoles et en permettant une personnalisation avancée de la logique grâce à la définition de scénarios. Un plan de test décrit la séquence d'étapes que la solution Artillery.io exécutera lors du test de charge. Nous avons opté pour Artillery.io pour nos tests de charge en raison de sa légèreté, de sa flexibilité et de sa facilité de configuration. De plus, il nous offre la possibilité d'écrire des scénarios de test en utilisant le langage de balisage YAML (Yet Another Markup Language), ce qui les rend faciles à lire et, dans la plupart des cas, plus faciles à écrire.[1]

### 3.2.7 Installation et lancement de l'outil Artillery.io :

Il est important de souligner qu'Artillery.io n'est pas destiné à tester des applications Web côté client via un navigateur ou JavaScript. Il est spécifiquement conçu pour tester les systèmes back-end tels que les APIs.

L'installation d'Artillery.io est relativement simple. Tout d'abord, assurez-vous d'avoir un système d'exploitation exécutant Node.js version 12.0 ou supérieure. Une fois que Node.js est configuré sur votre système, l'installation d'Artillery.io peut être effectuée en une seule ligne de commande : **npm install -g artillery**

Avant de commencer à utiliser Artillery, il est impératif de créer un script de test. Artillery est un outil qui fonctionne en exécutant un script au format YAML ou JSON. Ces scripts de test sont composés de deux sections essentielles : la section de configuration du test, appelée "config", et la section des scénarios à exécuter pour chaque utilisateur virtuel.

La section de configuration du test permet de définir minutieusement les paramètres essentiels tels que les cibles à tester, les paramètres de charge, les métriques à collecter, etc. Cette section offre une spécification précise à Artillery sur où et quoi tester, tout en permettant de déterminer le nombre d'utilisateurs virtuels à générer.

Quant à la section des scénarios, elle est dédiée à la spécification des interactions et des actions à effectuer pour chaque utilisateur virtuel pendant le test.

Les développeurs peuvent y définir des requêtes HTTP, des connexions WebSocket, des assertions de réponse, et d'autres actions nécessaires pour simuler le comportement réel des utilisateurs et évaluer les performances du système dans divers scénarios.[1]

En utilisant ces deux sections, il est possible de décrire de manière précise les objectifs et les paramètres du test de charge avec Artillery.

L'exemple de code présenté dans listing 1 est un script simple. Ce script se connecte à un point d'entrée d'API spécifique pendant une minute, en envoyant une charge de 10 utilisateurs virtuels par seconde.

config :

```
target : "http://127.0.0.1:4000"
```

phases :

```
- name : testpost
```

```
  duration : 60
```

```
  arrivalRate : 10
```

### Listing 1 : Code YAML définissant les phases

Comme on avait mentionnées précédemment, les scripts de test Artillery sont composés de deux sections principales. La première section, "config", contient deux paramètres de configuration : "target" qui spécifie l'URL de base pour toutes les requêtes définies ultérieurement dans le script de test, et "phases" qui précise la durée du test, le nombre d'utilisateurs virtuels que nous souhaitons envoyer pendant le test.

Le paramètre "scenarios" permet de spécifier une ou plusieurs requêtes que le programme Artillery exécutera pendant le test. Chaque scénario est une séquence d'étapes qui s'exécutent dans l'ordre, tel que défini par la clé "scenarios".

La clé "scenarios" contient chaque étape du scénario à exécuter. Étant donné que nous testons une API HTTP sous charge, nous utiliserons des méthodes de requête HTTP telles que GET et POST pour indiquer à Artillery.io les actions à effectuer tout au long du processus. Dans notre exemple, nous avons un seul scénario qui consiste en une requête GET, une utilisation de MongoDB, et une utilisation de PostgreSQL afin de comparer les performances.

Pour lancer la simulation, nous exécutons la commande suivante : `artillery run -e dev test.yml`

À la fin du test, Artillery.io rassemble les résultats de l'ensemble du test :

All VUs finished. Total time: 55 seconds

-----  
Summary report @ 23:33:36(+0200)  
-----

```
http.codes.200: ..... 600
http.request_rate: ..... 10/sec
http.requests: ..... 600
http.response_time:
  min: ..... 1
  max: ..... 85
  median: ..... 3
  p95: ..... 5
  p99: ..... 7
http.responses: ..... 600
vusers.completed: ..... 600
vusers.created: ..... 600
vusers.created_by_name.PostgresSql Query: ..... 600
vusers.failed: ..... 0
```

### Listing 2 : Résultat de la commande run de Artillery

Les résultats obtenus grâce à Artillery fournissent les informations essentielles pour évaluer le comportement du système. Artillery fournit des statistiques telles que le nombre moyen de réponses par seconde, les temps de réponse minimum, maximum, médian et les centiles (P90, P95). Ces données nous ont été particulièrement utiles pour mener notre étude comparative entre les bases de données SQL et NoSQL. Elles nous ont permis d'analyser les performances des deux types de bases de données et d'identifier les différences de comportement sous différentes charges de travail.

#### 3.2.8 DBeaver :

DBeaver est un logiciel open source qui supporte différents types de bases de données tel que (MySQL, SQLite, Oracle, PostgreSQL, SQL Server, MariaDB...).

Il prend en charge les systèmes d'exploitation Windows, Mac et Linux.

- La table en dessous présente la base de données SQL qu'on a crée en DBeaver, elle décrit un exemple d'un TP électronique , où les colonnes présente l'"id", le "temps" en millisecondes, la "tension" en Ampère ainsi le "courant" en voltage, sélectionnés à partir de la table mesure.[11]

select * from mesure			
id	temps	tension	courant
1	24Ms	42A	14V
2	33Ms	25A	23V
3	42Ms	22A	0V
4	7Ms	11A	3V
5	8Ms	7A	34V
6	34Ms	34A	0V
7	12Ms	4A	59V
8	35Ms	54A	24V
9	46Ms	56A	46V
10	38Ms	35A	4V
11	53Ms	51A	19V
12	2Ms	33A	21V
13	1Ms	15A	14V
14	40Ms	51A	4V
15	11Ms	49A	46V
16	43Ms	52A	7V
17	41Ms	39A	39V
18	40Ms	40A	23V

Figure 27 : Table SQL crée en DBeaver

### 3.2.9 MongoDB :

MongoDB est un système de base de données NoSQL qui gère un grand volume de données , se repose sur une structure en schéma dynamique orientée documents.[12]

mesures				
_id	ObjectId	Temps	Tension	Courant
1	ObjectId('64764f6c4c3fbf5a5a5...)	"12Ms"	"9 V"	"36 A"
2	ObjectId('64764f6c4c3fbf5a5a5...)	"41Ms"	"24 V"	"39 A"
3	ObjectId('64764f6c4c3fbf5a5a5...)	"30Ms"	"37 V"	"14 A"
4	ObjectId('64764f6c4c3fbf5a5a5...)	"39Ms"	"54 V"	"47 A"
5	ObjectId('64764f6c4c3fbf5a5a5...)	"20Ms"	"49 V"	"38 A"
6	ObjectId('64764f6c4c3fbf5a5a5...)	"12Ms"	"45 V"	"41 A"
7	ObjectId('64764f6c4c3fbf5a5a5...)	"58Ms"	"45 V"	"5 A"
8	ObjectId('64764f6c4c3fbf5a5a5...)	"12Ms"	"3 V"	"56 A"
9	ObjectId('64764f6c4c3fbf5a5a5...)	"30Ms"	"13 V"	"14 A"
10	ObjectId('64764f6c4c3fbf5a5a5...)	"33Ms"	"55 V"	"1 A"

Figure 28 : Table NoSQL créer en MongoDB Compass

### 3.3 Le Big Data :

Dans MongoDB, une base de données NoSQL, le Big Data peut être géré efficacement grâce à sa capacité à gérer des volumes massifs de données non structurées. MongoDB offre une grande flexibilité en termes de schéma, ce qui facilite l'ajout et la modification des données. De plus, MongoDB peut être distribué sur plusieurs serveurs pour permettre une meilleure évolutivité horizontale.

D'un autre côté, PostgreSQL, une base de données relationnelle, est également adapté pour le Big Data. PostgreSQL offre des fonctionnalités avancées telles que la réplication, la partitionnement des tables et l'optimisation des requêtes, ce qui le rend capable de gérer efficacement de grandes quantités de données.[17]

### 3.4 Langages, Frameworks, Plateformes :

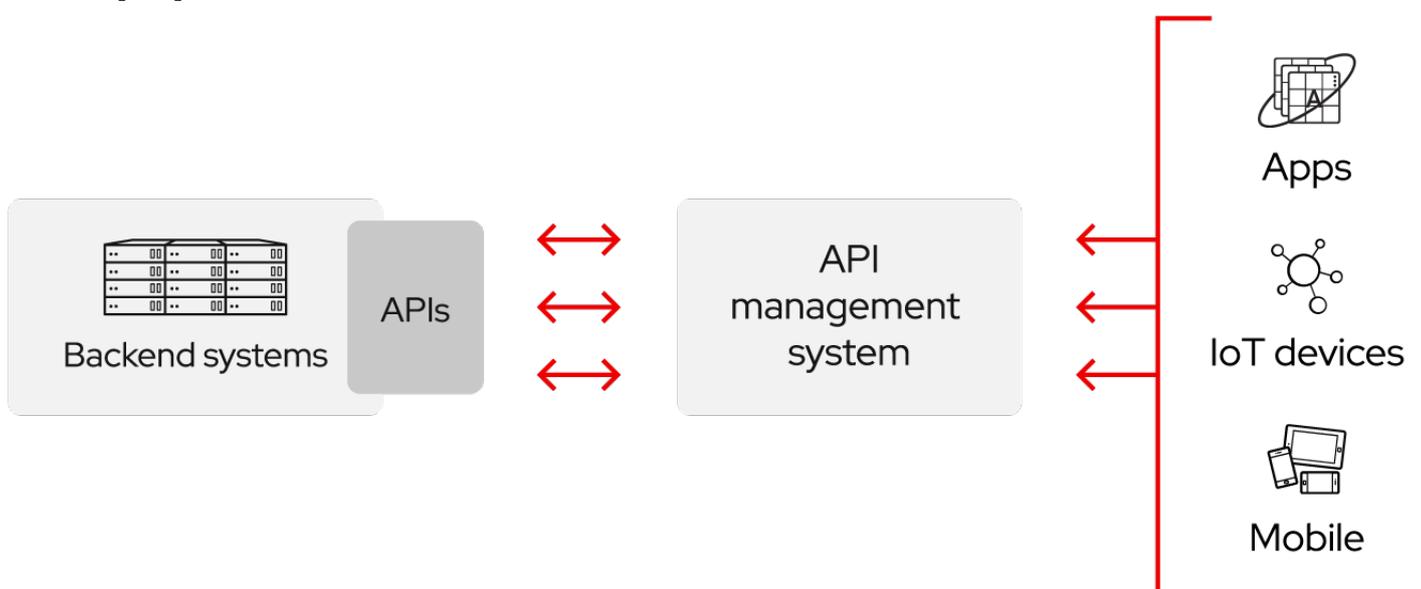
#### 3.4.1 API :

Une API (Application Programming Interface) est un ensemble de règles et de protocoles qui permettent à des logiciels de communiquer entre eux. Elle définit les méthodes et les formats de données utilisés pour l'échange d'informations entre différentes applications, services ou systèmes informatiques.

L'API sert de pont entre les différentes parties d'un système, en exposant certaines fonctionnalités et données spécifiques de manière sécurisée et contrôlée. Elle définit les points d'entrée (endpoints) par lesquels les développeurs peuvent accéder aux fonctionnalités offertes par une application ou un service.

Une API peut être conçue pour diverses utilisations, telles que la récupération de données à partir d'une base de données, la manipulation de fichiers, l'accès à des fonctionnalités spécifiques d'un logiciel, l'intégration de services tiers, etc. Les API peuvent prendre différentes formes, telles que des API Web basées sur les protocoles HTTP/HTTPS, des API de bibliothèques logicielles, des API de bases de données, des API de services cloud, etc.

En résumé, une API est un ensemble de règles et d'interfaces standardisées qui permettent aux développeurs d'utiliser les fonctionnalités d'une application ou d'un service sans avoir à connaître les détails de sa mise en œuvre interne.[15]



## Figure 29 : interactions de l'API[15]

### 3.4.2 JavaScript :

JavaScript est un langage de programmation utilisé principalement dans les navigateurs web pour rendre les pages interactives et dynamiques. Toutefois, JavaScript est également utilisé en dehors des navigateurs dans d'autres contextes, tels que les bases de données et les API.

JavaScript côté serveur (ou JavaScript côté serveur) est utilisé dans les bases de données pour exécuter des scripts et des opérations sur les données stockées. Cela permet de manipuler et de gérer les données de manière dynamique. Les bases de données modernes, comme MongoDB, permettent l'utilisation de JavaScript côté serveur pour effectuer des opérations complexes, des agrégations de données et des mises à jour en utilisant JavaScript.

JavaScript est également utilisé dans le contexte des API (Application Programming Interface). Les API permettent à différents logiciels et services de communiquer et d'échanger des données entre eux. JavaScript est souvent utilisé pour consommer et interagir avec les API, en envoyant des requêtes HTTP pour récupérer des données à partir de l'API et en manipulant les réponses pour les afficher ou les utiliser dans une application web.

En résumé, JavaScript est utilisé en base de données pour exécuter des scripts et manipuler les données côté serveur, et il est utilisé dans les API pour interagir avec les services et récupérer des données via des requêtes HTTP.[16]

### 3.4.3 YAML :

Acronyme de "Yet Another Markup Language" dans sa version 1.0 et "YAML Ain't Markup Language" dans sa version 1.1.

YAML n'est pas un langage de balisage, il est un format de sérialisation de données souvent utilisé pour configurer des applications ou échanger des données entre différents systèmes.

Le format YAML se distingue par son utilisation de l'indentation et des caractères spéciaux pour structurer les informations de manière hiérarchique. Il est populaire dans de nombreux domaines, tels que les configurations de logiciels, les fichiers de définition d'infrastructure et les flux de données pour les applications web et mobiles. YAML est souvent préféré pour sa simplicité, sa lisibilité et son support multi-langages.[13]

### 3.4.4 Node js :

- Node.js est une plateforme logicielle libre qui utilise JavaScript sur le serveur.
- Node.js est un environnement de serveur open source.
- Node.js est gratuit.
- Node.js fonctionne sur différentes plates-formes (Windows, Linux, Unix, Mac OS X, etc.)[14]

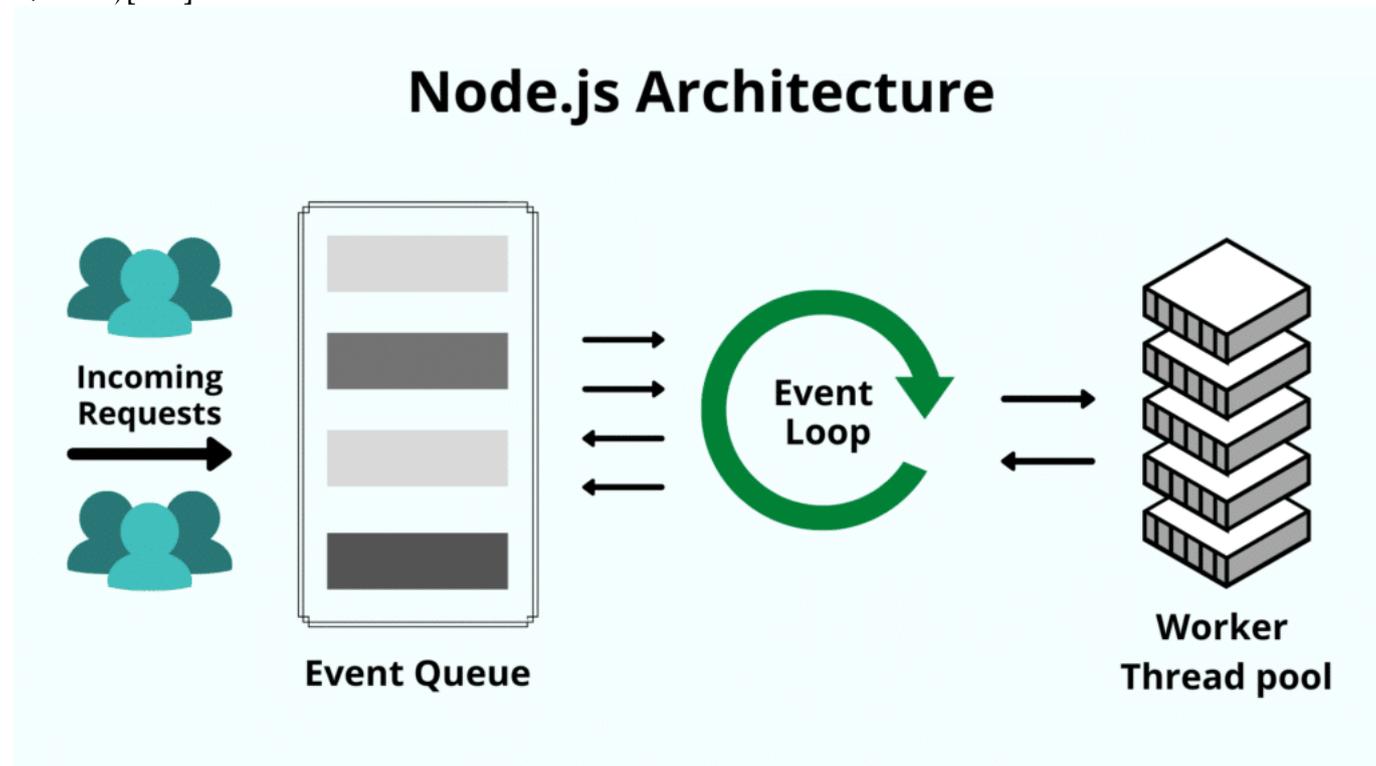


Figure 30 : Node.js architecture[29]

### 3.4.5 Nodemon :

- Nodemon est un package qui met à jour automatiquement le serveur à chaque modification de fichier de manière continue, ce qui facilite le développement Node.js. Contrairement à Node.js, qui nécessite de suivre manuellement les mises à jour du serveur à chaque changement effectué, avec Nodemon, vous pouvez simplement accéder au bon chemin du serveur et exécuter la commande suivante : '`nodemon index.js`'. Cela permettra au serveur de se mettre à jour automatiquement à chaque modification de fichier. [17]

### 3.4.6 Express js :

Express.js est un framework populaire pour le développement d'applications web utilisant Node.js, un environnement d'exécution JavaScript. Il a été conçu pour simplifier le processus de création d'applications et d'API web en fournissant un ensemble d'outils flexibles et minimalistes.

Express.js suit l'architecture des middlewares, permettant aux développeurs de créer des composants modulaires et réutilisables pour gérer des tâches spécifiques dans le flux d'application. Il propose diverses fonctionnalités pour le routage, la gestion des requêtes et réponses HTTP, la gestion des sessions, et bien plus encore.

Avec Express.js, vous pouvez définir des routes pour gérer différentes URL et méthodes HTTP, spécifier des fonctions middleware pour traiter les requêtes et réponses, et intégrer différents plugins ou modules pour étendre ses capacités. Il fournit également une API simple et intuitive pour interagir avec les bases de données, facilitant ainsi la création d'applications web basées sur des bases de données.

Dans l'ensemble, Express.js est reconnu pour sa simplicité, sa flexibilité et sa scalabilité, ce qui en fait un choix populaire parmi les développeurs pour la création d'applications web et d'API en utilisant Node.js.[19]

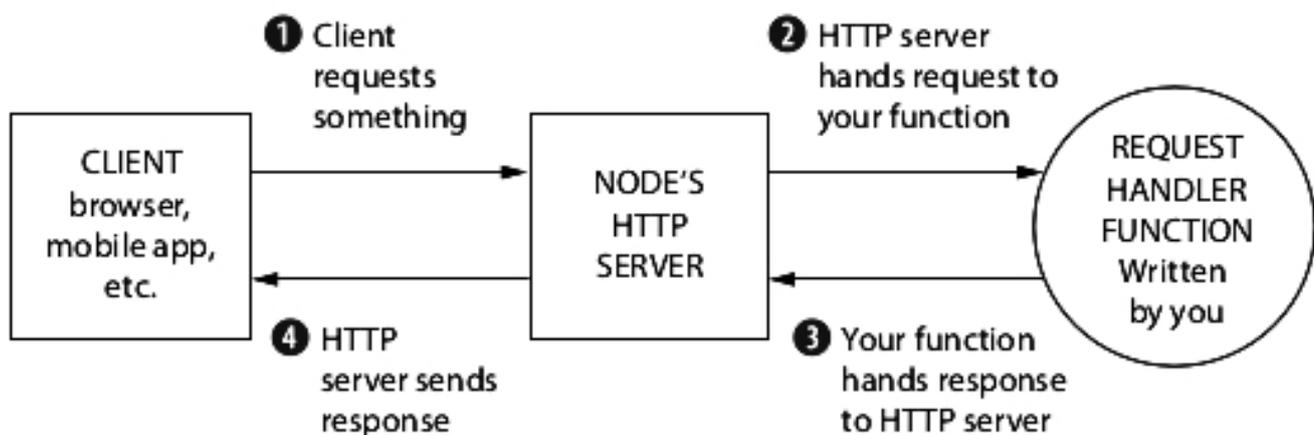


Figure 31 : Node HTTP serveur[30]

## 3.5 Évaluation de la performance dans l'environnement NoSql et SQL :

### 3.5.1 1- Dans l'environnement NoSQL :

Dans le cadre de notre étude portant sur l'évaluation des performances des bases de données NoSQL, notre objectif principal était de parvenir à un délai de réponse de 5 secondes.

Voici le script YAML que nous avons utilisé pour exécuter notre programme et obtenir les résultats que nous cherchions :

```
config :
  target : "http://127.0.0.1:4000"
  phases :
    - name : testpost
      duration : 60
      arrivalRate : 10
  scenarios :
    - name : MongoDB
      - get :
        url : /api/nosql/mesures?measure=$randomNumber(1,1000000)}
```

**Listing 3 :** Code YAML définissant les phases et scénarios pour NoSQL.

- "**testpost**" a une durée de 60 secondes, ce qui signifie que le test de charge sera exécuté pendant une minute.
- "**arrivalRate**" est défini sur 10, ce qui indique que 10 requêtes seront envoyées par seconde pendant la phase de test. Cela contrôle le débit des requêtes envoyées au serveur.
- Le scénario du test est nommé "**Mongodb**".
- La requête GET est envoyée à l'URL `"/api/nosql/mesures"` avec un paramètre `"measure"` généré aléatoirement entre 1 et 1 000 000.

Le tableau ci-dessous présente en détail les différentes expérimentations que nous avons réalisées pour atteindre cet objectif :

UV	Temps(s)	P95	P99
10	60	1002.4	1002.4
100	60	1495.5	1686.1
120	300	1153.1	1353.1
150	300	1249.1	1465.9
200	300	1274.3	1863.5
400	300	8692.8	9801.2
400	900	1002.4	1107.9
800	900	1587.9	1901.1
900	900	1556.5	2369.0
1000	900	3465.9	4755.0
1100	900	6976.1	9230.4
1500	1800	2231.0	3134.5
2000	1800	3328.3	4065.2
2100	1800	31345.5	6976.1
2500	3600	1022.7	1300.1
3500	3600	1587.9	3328.3
4000	3600	1686.1	5272.4

**Tableau02** : Tableau d'évaluation des performances pour atteindre un temps de réponse de 5s dans NoSQL

**UV** : utilisateurs virtuels.

**Temps** : en secondes.

**P95** : 95 pourcent des utilisateurs réussi à faire une requête en millisecondes.

**P99** : 99 pourcent des utilisateurs réussi à faire une requête en millisecondes.

Nous avons lancé l'étude en utilisant une base de données contenant 1 million d'enregistrements. Au départ, nous avons démarré avec 10 utilisateurs en seulement 60 secondes à l'aide de l'outil d'Artillery. Au fil du temps, nous avons progressivement augmenté le nombre d'utilisateurs pour atteindre finalement 4000 utilisateurs en une heure. Cette évolution nous a permis d'atteindre notre objectif fixé de temps de réponse avec un délai de 5 secondes pour 95 à 99 pourcent des utilisateurs ayant effectué une requête.

### 3.5.2 2- Dans l'environnement SQL :

Notre deuxième étude était axée sur l'évaluation des performances dans le domaine SQL. L'objectif principal était d'obtenir un temps de réponse de 5 secondes. Nous avons effectué plusieurs expérimentations détaillées ci-dessous, dans le but de parvenir à cet objectif :

UV	Temps(s)	P95	P99
10	60	3	7
100	60	5	7
120	300	4	6
150	300	4	5
200	300	4	5
400	900	4	5
800	900	4	6
1000	900	4	5
1500	1800	3	5
2000	1800	3	5
2100	1800	3	5
4000	3600	3	5

**Tableau03** : Tableau d'évaluation des performances pour atteindre un temps de réponse de 5s dans SQL

**UV** : utilisateurs virtuels.

**Temps** : en secondes.

**P95** : 95 pourcent des utilisateurs réussi à faire une requête en millisecondes.

**P99** : 99 pourcent des utilisateurs réussi à faire une requête en millisecondes.

Voici le script YAML que nous avons utilisé pour exécuter notre programme et obtenir les résultats que nous cherchions :

```
config :
  target : "http://127.0.0.1 :4000"
  phases :
    - name : testpost
      duration : 60
      arrivalRate : 10
  scenarios :
    - name :PostgresSql Query
      - get :
          url: /api/sql/mesure?id={{$randomNumber(1,1000000)}}$
```

**Listing 4** : Code YAML définissant les phases et scénarios pour SQL.

- **"testpost"** a une durée de 60 secondes, ce qui signifie que le test de charge sera exécuté pendant une minute.
- **"arrivalRate"** est défini sur 10, ce qui indique que 10 requêtes seront envoyées par seconde pendant la phase de test. Cela contrôle le débit des requêtes envoyées au serveur.
- Le scénario du test est nommé **"PostgresSql Query"**.
- La requête GET est envoyée à l'URL **"/api/sql/mesures"** avec un paramètre **"id"** généré aléatoirement entre 1 et 1 000 000.

Initialement, notre étude a débuté en utilisant une base de données contenant 1 million d'enregistrements, qui était également utilisée dans le cadre de notre étude sur NoSQL. Au fil de notre étude, nous avons progressivement augmenté le nombre d'utilisateurs jusqu'à atteindre 4000 utilisateurs tout comme dans l'étude de NoSQL. Malgré cela, le temps de réponse est resté considérablement réduit par rapport à NoSQL. .

Par la suite, nous avons relevé un défi majeur en augmentant le volume de données à 5 millions d'enregistrements, Le tableau ci-dessus présente les résultats obtenus :

UV	Temps(s)	P95	P99	UR
10	60	6	6	
100	60	4	6	
1000	60	2	3	
100000	120	5	13	49083
100000	300	3	7	50000
100000	900	2	3	
200000	900	2	5	121581
200000	1800	2	4	
300000	3600	2	4	
500000	3600	3	5	475369

**Tableau04** : Tableau d'évaluation des performances pour atteindre un temps de réponse de 5s dans SQL avec 5M de data

**UV** : utilisateurs virtuels.

**Temps** : en secondes.

**P95** : 95 pourcent des utilisateurs réussi à faire une requête en millisecondes.

**P99** : 99 pourcent des utilisateurs réussi à faire une requête en millisecondes.

**UR** : Utilisateurs virtuels qui ont réussi à effectuer une requête et à obtenir une réponse.

- Nos observations ont révélé que notre solution maintient un temps de réponse réduit par rapport à NoSQL, ce qui témoigne de sa performance supérieure.

Une fois de plus, nous avons relevé le défi avec SQL. Nous avons reconfiguré le script YAML en y ajoutant trois phases, comme vous pouvez le constater dans le code présenté ci-dessous :

phases :

- name : testpost  
duration : 60  
arrivalRate : 10
- name : testpost2  
duration : 300  
arrivalRate : 100  
rampTo : 10
- name : testpost3  
duration : 900  
arrivalRate : 100

### Listing 5 :Code YAML définissant les phases.

Voici le déroulement des phases et le resultat fourni par Artillery :

- Phase 1 : 10 utilisateurs par seconde pendant 60 secondes.
- Phase 2 : 100 utilisateurs par seconde pendant 300 secondes avec ramp de 10 utilisateurs
- Phase 3 : 100 utilisateurs par seconde pendant 900 secondes.

All VUs finished. Total time: 21 minutes , 4 seconds

-----  
Summary report @ 12:03:36(+0200)  
-----

```
http.codes.200: ..... 107100
http.request_rate: ..... 85/sec
http.requests: ..... 107100
http.response_time:
  min: ..... 0
  max: ..... // ..... 876
  median: ..... 2
  p95 : ..... 4
  p99: ..... 12.1
http.responses: ..... 107100
vusers.completed: ..... 107100
vusers.created: ..... 107100
vusers.created_by_name.PostgresSql Query: ..... 107100
vusers.failed: ..... 0
```

### Listing 6 : Résultat de l'étude avec Artillery

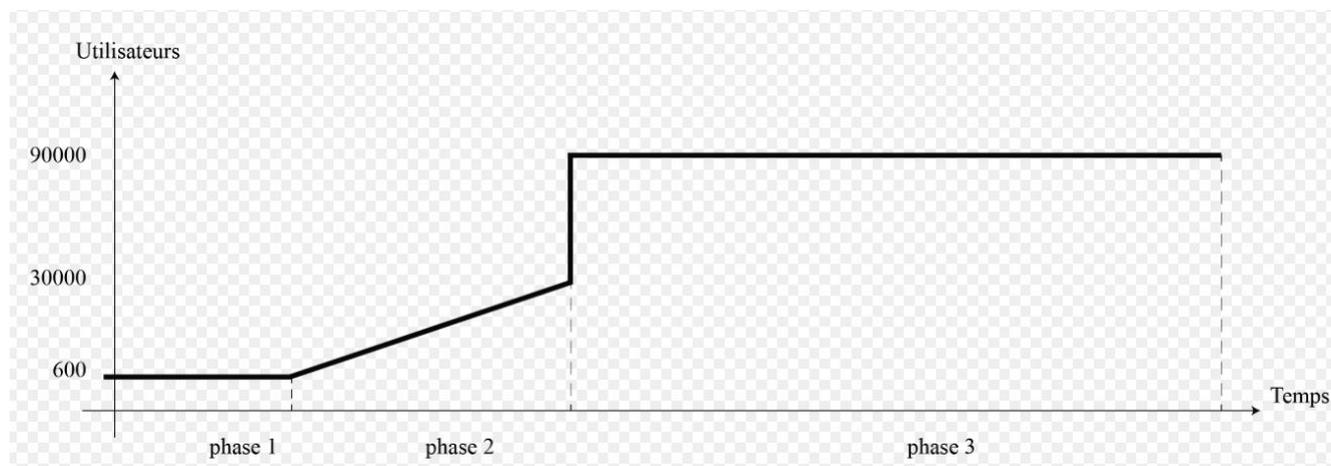


Figure 32 : l'ensemble des phases de Artillery.io

Les résultats indiquent qu'au cours d'une période de 21 minutes, nous avons enregistré l'entrée de 107 100 utilisateurs. De plus, 99 pourcent de ces utilisateurs ont réussi à effectuer une requête avec un temps de réponse de 12,1 ms.

### **3.6 Conclusion :**

D'après l'étude réalisée, le SQL se révèle être préférable au NoSQL dans plusieurs cas pour l'étude du Big Data. Il permet une manipulation et une analyse efficaces des données, facilitant ainsi l'obtention d'informations précises et exploitable dans un temps réduit.

# Conclusion

Ce mémoire a examiné en détail le concept de Remote Lab à travers trois chapitres distincts. Le premier chapitre a fourni une définition approfondie du Remote Lab, soulignant ses principales caractéristiques et avantages. Nous avons constaté que le Remote Lab est une solution innovante qui permet un accès à distance à des équipements de laboratoire, des appareils ou des expériences, éliminant ainsi les contraintes géographiques et temporelles.

Le deuxième chapitre a présenté différents projets de Remote Lab dans divers domaines. Nous avons étudié des exemples concrets de Remote Lab dans les domaines de l'éducation, de la recherche scientifique et de l'industrie. Ces projets ont démontré l'ampleur des possibilités offertes par le Remote Lab et son potentiel à transformer les méthodes traditionnelles de laboratoire.

Le troisième chapitre s'est concentré sur une étude comparative entre les bases de données SQL et NoSQL dans le contexte du Remote Lab. Nous avons analysé les caractéristiques, les avantages et les limites de ces deux types de bases de données, en évaluant leur pertinence pour les applications du Remote Lab. Au cours de cette étude comparative, nous avons conclu que SQL présente des avantages tels que la cohérence des données, la gestion des transactions et la flexibilité des requêtes. Cependant, nous avons également constaté que le choix entre SQL et NoSQL dépend des besoins spécifiques du Remote Lab. Dans certains cas, NoSQL peut être préféré en raison de sa scalabilité horizontale, de sa capacité à gérer de grands volumes de données non structurées et de sa flexibilité pour les modèles de données évolutifs.

En résumé, ce mémoire a approfondi notre compréhension du Remote Lab et de son potentiel d'application dans divers domaines. Les résultats de cette recherche soulignent l'importance de considérer les besoins spécifiques du Remote Lab lors du choix de la base de données, que ce soit SQL ou NoSQL. Chaque système présente ses propres avantages et limitations, et il est essentiel de prendre en compte les exigences de l'application et les caractéristiques des données pour prendre une décision éclairée.

Annexes :

## Listing 7 : Code YAML définissant les phases et scenarios

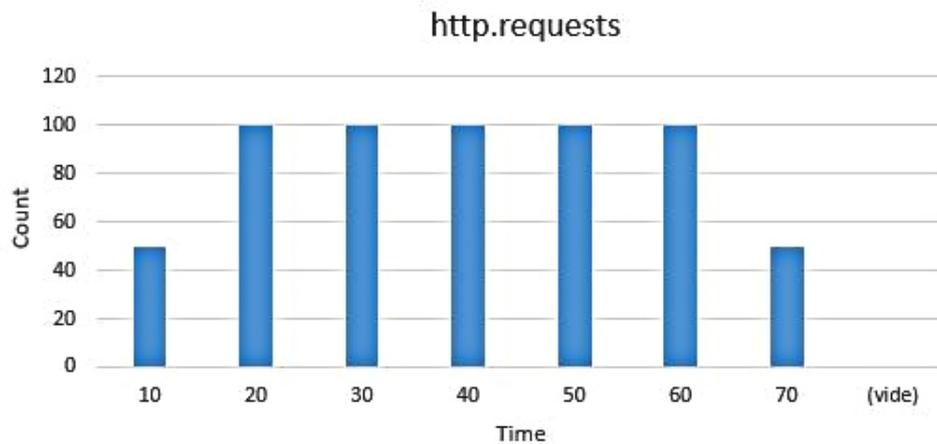
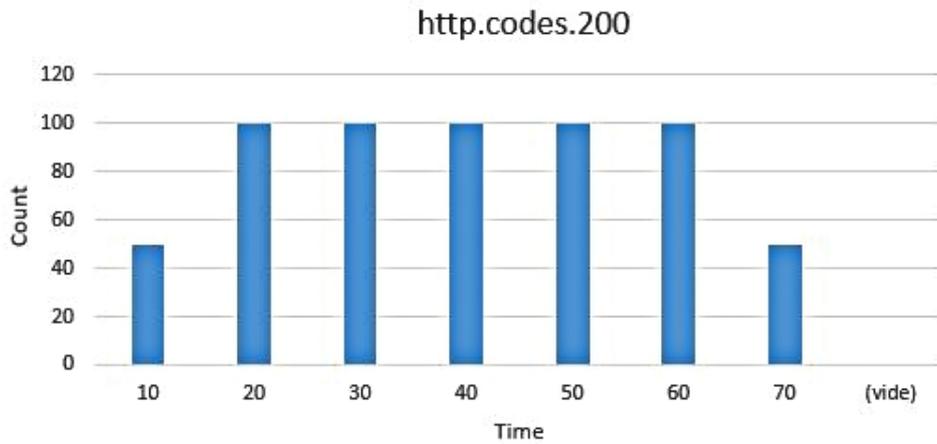
```
config:
  environments:
    dev:
      target: "http://127.0.0.1:4000"
      phases:
        - name: testpost
          duration: 60
          arrivalRate: 10
        - name: testpost2
          duration: 300
          arrivalRate: 10
          rampTo: 100
        - name: testpost3
          duration: 900
          arrivalRate: 100

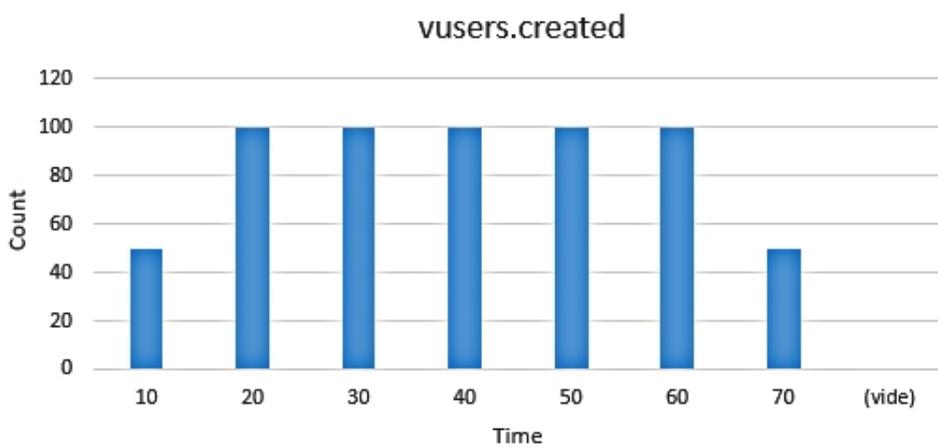
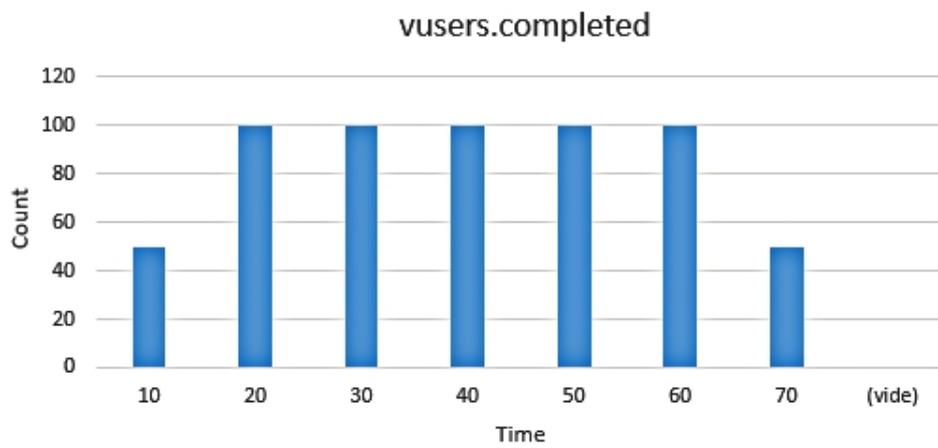
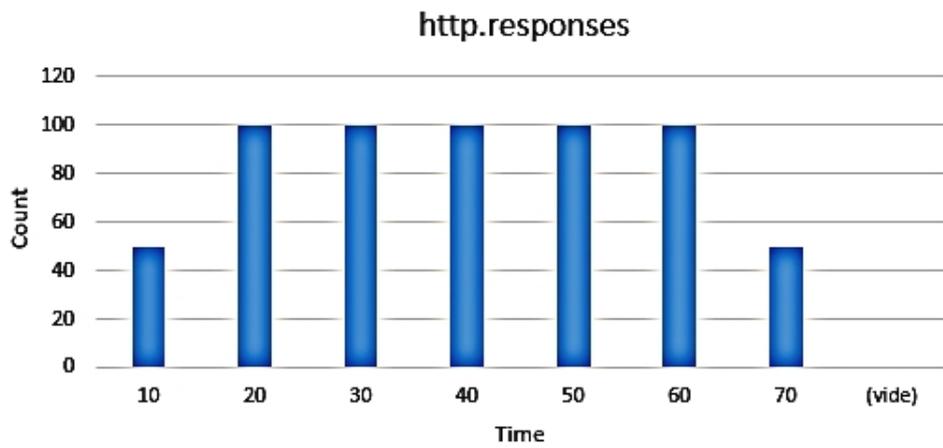
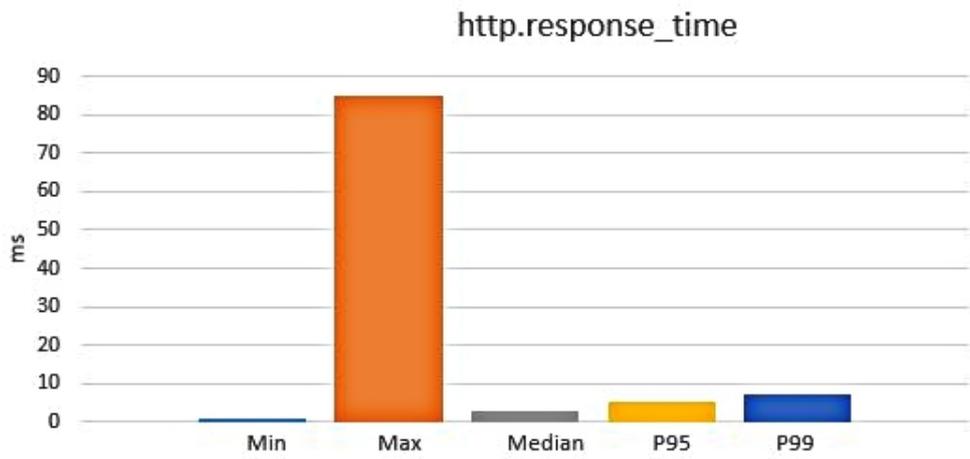
  scenarios:
    - name: PostgresSql Query
      flow:
        - get:
            url: /api/sql/product?id={{$randomNumber(1,1000000)}}

    - name: Mongoddb
      flow:
        - get:
            url: /api/nosql/product?product={{$randomNumber(1,1000000)}}
```

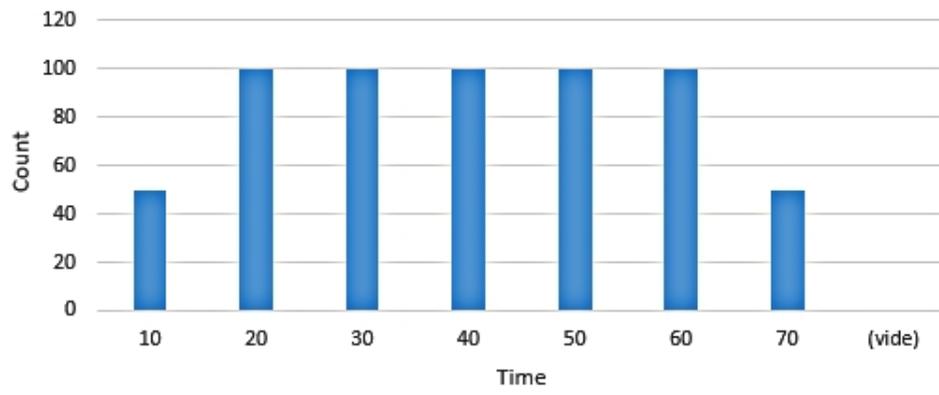
## Artillery report

<b>Duration</b>	0m 59s
<b>Started</b>	14/06/2023, 10:18:14
<b>Completed</b>	14/06/2023, 10:19:14
<b>Errors</b>	0
<b>Metric</b>	<b>Value</b>
http.codes.200	600
http.requests	600
http.responses	600
vusers.completed	600
vusers.created	600
vusers.created_by_name.PostgresSql Query	600
vusers.failed	0





vusers.created\_by\_name.PostgresSql Query



# Bibliographie :

## 3.6.1 Thèse Doctorat :

[1] : Thèse Doctorat Mohammed Moussa-2021 : " Conception d'un environnement d'apprentissage collaboratif interopérable dédié aux travaux pratique distants" e-biblio.univ-mosta.dz"

[31] : X.Chenet al. « Virtual and remote laboratory development : A review ».Earth and Space 2010 : Engineering, Science, Construction, andOperationsin Challenging Environments, 2010

[32] : J. Umborg et A. Uukkivi. « Continuity in the Development of Technical Thinking ». In : Developing Technology Mediationin Learning Environments. IGI Global, 2020

## 3.6.2 Google Scholar :

[6] : 2020 IEEE Frontiers in Education Conference (FIE) The VISIR project - an Open Source Software Initiative for Distributed Online Laboratories I. Gustavsson, J. Zackrisson, +2 authors T. Lagö Published 2007 Computer Science -Google Scholar-VISIR-

[7] : Remote Experimentation with Massively Scalable Online LaboratoriesRemote Experimentation with Massively Scalable Online Laboratories January 2018 DOI :10.1007/978-3-319-64352-6\_24 In book : Online Engineering Internet of Things (pp.258-265) Authors : Lars Thorben Neustock Stanford University George Herring Stanford University Lambertus Hesselink Stanford University -Google Scholar-

## 3.6.3 Youtube :

[3] : Radhi Mhiri "LAD2" 6 avr. 2020 Enseignement à distance : La question de travaux de laboratoire ( TL) pendant la pandémie! -Youtube-

[4] : Radhi Mhiri "LAD2 VS UQO" Sortie le 16 mars 2022 Description de l'expérience de laboratoire à distance basée sur l'utilisation de Kit de TP (UQO) -Youtube-

### 3.6.4 PDF :

[18] : BigData\_Technologies\_PL.pdf -BigData-  
A Survey on Big Data Technologies and Their Applications to the Metaverse : Past,  
Current and Future January 2023 Authors : Hao Lan Zhang NIT, Zhejiang University  
and Sanghyuk Lee Xi'an Jiaotong-Liverpool University

### 3.6.5 Article :

[20]17 mai 2001 - Journée mondiale des télécommunications L'Internet : enjeux,  
opportunités et perspectives  
Figure 2 : Croissance et inégalités  
Serveurs Internet installés, janvier 1991 - juillet 1999, et répartition par région,  
juillet 1999 -itu.int-

### 3.6.6 website :

[2] : What is IoT? "oracle.com"

[5] : Laboratoire à faible coût pour l'enseignement pratique à distance Najib  
Abekiri Dans Projectics / Proyética / Projectique 2022/3 (n°33), pages 65 à 73  
"www.cairn.info"

[9] : Overleaf La dernière modification de cette page a été faite le 6 mars 2023  
à 12 :02 "Wikipedia.com"

[10] : overleaf "fr.overleaf.com"

[11] : Telecharger DBeaver "clubic.com"

[12] : MongoDB : Learn all about the NoSQL database May 6, 2021 "datascien-  
tist.com"

[14] : Qu'est-ce que Node.js et pourquoi l'utiliser? Dernière mise à jour : 31 mai  
2023 Architecture de Node.js et comment elle fonctionne Comment node.js traite  
les requêtes entrantes en utilisant la boucle d'événement "kinsta.com"

- [15] : What's an API Published June 2, 2022 "RedHat.com"
- [16] : ACTUS Guide complet pour apprendre JavaScript REMY LARQUETOUX 9 JUIN, 2023 INFORMATIQUE, DÉVELOPPEUR "ETUDESTECH.com"
- [17] : Nodemon Vs.Node by S Shariff Sep 13, 2021 "medium.com"
- [19] : What Is Express.js? Everything You Should Know "kinsta.com"
- [21] : How to Use an Oscilloscope "Sparkfun.com"
- [22] : What is the Internet of Things (IoT)? "TIBC.com"
- [24] : ÉLECTRONIQUE PROGRAMMABLE ET ROBOTIQUE by Julien Naud Mostert Installateur chez optel vision "cegeplimoilou.ca"
- [25] : laboratoire virtuel "grenlog.free.fr"
- [26] : What is Web Application ? By Mahantesh NagathanMahantesh Nagathan "deducba.com"
- [27] : man-scientifique-en-blouse-blanche-joue-flacon-avec-notre-nouvelle-médecine "123rf.com"
- [2] : - Figure 3 - uploaded by Remi Venant-1  
Topologie de l'architecture iLab pour des expériences interactives (Harward et al., 2008)  
"researchgate.net"
- [23] : SQL vs NoSQL : What's the Difference ? Posted on April 24, 2021 by Mike Shannon sql-vs-nosql "gowithcode.com"
- [28] : "https://code.visualstudio.com/"
- [29] : [Node.js] Securing your applications Part 3/3 by : Marian Villa in Community on Jan 19 2022 Node.js Security — Architecture "nodesource.com"
- [8] : Mettez en place votre environnement front-end  
Explorez les fonctionnalités de Visual Studio Code  
Le professeur Will Alexander  
Scottish developer, teacher and musician based in Paris."openclassrooms"

[13]: YAML, qu'est-ce que c'est ? Publié 3 mars 2023 [https://www.redhat.com/fr/topics-is-yaml](https://www.redhat.com/fr/topics/is-yaml) "redhat.com"

[30]: Créer un serveur avec Express.js Express.js "cril.univ-artois.fr"