

## **DEDICACE**

**A toute la famille SARWAL**

## REMERCIEMENTS

Ce document témoigne de l'achèvement d'un long processus d'études que j'ai entrepris il y a deux ans déjà. Au cours de cette aventure, la chance m'a été donnée de côtoyer nombre de gens qui ont eu, directement ou indirectement, une influence positive sur l'accomplissement de ce projet. Ils ont été mes formateurs, mes amis, mes conseillers, ma famille et chacun d'eux, à leur manière, a contribué à la réalisation de ce mémoire.

Je souhaite remercier mes encadreurs, Dr Valéry MONTHE et M. Mathias NDJETEHE pour leur soutien et recommandations judicieuses. Ils ont su me guider et me donner les outils et les ressources nécessaires pour mener à terme ce projet. Je remercie également M. ENGOLO DELMAS Mérimée et M. NGONO MANGA Elysée pour leur gentillesse et le temps qu'ils m'ont consacré ainsi qu'à tout le personnel de la CAMTEL ayant contribué de près ou de loin dans le bon déroulement du projet.

Je dois chaque bribe de ma connaissance à mes enseignants de SUP'PTIC de Yaoundé et de SUPCOM de Tunis qui ont su bien mené leur noble quête d'enseigner tous ce qu'il faut. Je les remercie non seulement pour le savoir qu'ils m'ont transmis, mais aussi pour la fierté et l'ambition que leurs personnes m'aspirent.

Je remercie également les membres du jury pour leur présence, pour leur lecture attentive de mon mémoire ainsi que pour les remarques qu'ils m'adresseront lors de cette soutenance afin d'améliorer mon travail.

Non loin de tout projet ambitieux, il existe des gens qui partagent sans jamais se lasser nos meilleurs et pires moments. Ce sont des personnes qui m'ont soutenus dans chacun de mes instants de faiblesse, et qui, sans leur extrême attention et gentillesse, je ne serais pas ce que je suis devenu aujourd'hui. Je pense particulièrement à M. ADOUM MAHAMAT SARWAL pour son aide, son encouragement et sa disponibilité. Je pense également à tous mes camarades de promotion ainsi que tous mes amis qui m'ont aidés de façon directe ou tout simplement par leur présence et leur soutien moral.

## LISTE DES ABREVIATIONS

Termes	Significations
<b>AES</b>	Advanced Encrypted Standard
<b>CAMTEL</b>	Cameroon Telecommunications
<b>CBC</b>	Cipher Block Chaining
<b>CLI</b>	Command Line Interface
<b>Crocodile</b>	Cisco Router Configuration Diligent Evaluator
<b>CSS</b>	Cascading Style Sheets
<b>GASC</b>	Gestion automatisée et sécurisée des configurations
<b>HMTL</b>	HyperText Markup Language
<b>IETF</b>	Internet Engineering Task Force
<b>IHM</b>	Interface Homme Machine
<b>IP</b>	Internet Protocol
<b>ISO</b>	International Organization for Standardization
<b>LER</b>	Label Edge Router
<b>LSP</b>	Label Switched Path
<b>LSR</b>	Label Switched Router
<b>MPLS</b>	Multi Protocol Label Switching
<b>NetML</b>	Network Markup Language
<b>PHP</b>	HyperText PreProcessor
<b>RAT</b>	Router Audit Tool
<b>SGBD</b>	Système de Gestion des Bases de données
<b>SHA</b>	Secure Hash Algorithm
<b>SNMP</b>	Simple Network Management Protocol
<b>SQL</b>	Structured Query Language
<b>SSH</b>	Secure Shell
<b>TELNET</b>	Terminal Network
<b>UML</b>	Unified Modeling Language
<b>VNG</b>	Validated Network Generator
<b>XML</b>	Extensible Markup Language
<b>MERISE</b>	(Méthode d'Etude et de Réalisation)

	Informatique pour les Systèmes d'Entreprise)
<b>URL</b>	Unified Ressources Location
<b>IOS</b>	Internetwork Operating System

## RESUME

La gestion des réseaux informatiques est une tâche de plus en plus complexe et sujette aux erreurs. Les recherches dans le passé ont montré que la plupart des modifications apportées à la configuration d'un réseau échouent à leur première tentative d'utilisation, et la moitié de ces échecs sont motivés par un problème situé ailleurs dans le réseau. Les opérateurs des réseaux sont ainsi confrontés à un problème commun : comment s'assurer qu'un service installé sur le réseau d'un client fonctionne correctement et que le réseau lui-même est exempt de défaut de toute nature ? L'ingénieur réseau a donc à chaque fois qu'un nouveau service sera ajouté au réseau, la responsabilité d'un groupe de périphériques dont les configurations sont gérées individuellement et manuellement.

L'évolution fulgurante du nombre de dispositifs, la complexité des configurations, les besoins spécifiques de chaque service, le nombre même de services qu'un réseau doit être capable de supporter, rendent cette tâche de plus en plus difficile. Nous pouvons aisément comprendre la nécessité de nouvelles approches au problème de gestion de configuration réseau.

Nous avons donc dans ce mémoire de fin d'études, proposé des approches qui ont abouties à la création d'une application web, écrite avec les dernières technologies du web intitulée *GASC*. Cette application est capable de générer les configurations réseaux pouvant être déployées sur les équipements physiques de l'opérateur. *GASC* permet également aux opérateurs de rapatrier les configurations fréquentes sur des équipements réseaux. *GASC* est unique, car elle laisse la possibilité aux opérateurs d'exécuter les commandes sur leurs équipements. Ils peuvent configurer simplement la liste des commandes qu'ils souhaitent appliquer à une catégorie de périphériques, puis créent une tâche planifiée et *GASC* fera le reste.

**Mot clés : Application web, gestion des configurations, PHP, MYSQL.**

## ABSTRACT

The management of computer networks is an increasingly complex and error-prone task. Past research has shown that most network configuration changes fail on their first attempt at use, and half of these failures are due to a problem elsewhere in the network. Network operators are thus faced with a common problem: how to ensure that a service installed on a customer's network works properly and that the network itself is free of defects of any kind? The network engineer is therefore responsible for a group of devices whose configurations are managed individually and manually each time a new service is added to the network.

The rapid evolution in the number of devices, the complexity of configurations, the specific needs of each service, the very number of services that a network must be able to support, make this task more and more difficult. We can easily understand the need for new approaches to the network configuration management problem.

In this thesis, we have therefore proposed approaches that have led to the creation of a web application, written with the latest web technologies called GASC. This application is able to generate network configurations that can be deployed on the operator's physical equipment. GASC also allows operators to repatriate frequent configurations on network equipment. GASC is unique because it allows operators to execute orders on their equipment. They can simply configure the list of commands they want to apply to a category of devices, then create a scheduled task and GASC will do the rest.

**Key words : Web application, configuration management, PHP, MYSQL.**

## TABLE DE MATIERES

DEDICACE.....	i
REMERCIEMENTS .....	ii
LISTE DES ABREVIATIONS .....	iii
RESUME.....	v
ABSTRACT .....	vi
TABLE DE MATIERES.....	vii
LISTE DES FIGURES.....	xi
LISTE DES TABLEAUX.....	xii
INTRODUCTION GENERALE.....	1
CHAPITRE 1 : CONTEXTE ET PROBLEMATIQUE .....	3
1.1 Contexte du travail .....	3
1.2 Problématique.....	3
1.2.1 Problèmes liés à la configuration des équipements réseaux.....	3
1.2.2 Problèmes liés à la sauvegarde des configurations réseaux.....	4
1.3 Objectifs.....	4
1.3.1 Objectif général .....	4
1.3.2 Objectifs spécifiques.....	4
1.4 Résultats Attendus .....	5
1.5 Méthodologie.....	5
1.6 Plan du mémoire.....	5
1.7 Etude de l'existant .....	6
1.8 Bilan critique de l'existant.....	6
1.9 Solution Proposée .....	7
1.10 Planning de réalisation du projet.....	8

CHAPITRE 2 : METHODOLOGIE DE LA CONCEPTION .....	10
I. ETATS DE L'ART .....	10
2.1 Gestion automatisée des configurations réseaux.....	10
2.1.1 Evénements de l'actualité.....	10
2.1.2 Les enjeux .....	10
2.1.3 Qu'est-ce qu'une configuration ?.....	11
2.1.4 Causes .....	12
2.1.5 Conséquences.....	12
2.2 Approches actuelles dans la gestion des configurations .....	12
2.2.1 Solutions standardisées.....	12
2.2.1.1 SNMP [6] .....	12
2.2.1.2 NetConf .....	13
2.2.2 Solutions non standardisées.....	13
2.2.2.1 RAT et Crocodile .....	13
2.2.2.2 NetML .....	14
2.2.2.3 MetaConfig.....	14
2.2.2.4 El-Arini et Killourhy .....	14
2.2.2.5 VNG .....	15
2.3 Etude comparative des solutions .....	15
2.3.1 Définition des critères de comparaison .....	15
2.3.2 Tableau comparatif.....	16
II. CONCEPTION DE LA SOLUTION PROPOSEE .....	18
3.1 Méthodes et langage d'analyse.....	18
3.1.1 Langage de modélisation UML .....	18
3.1.2 Win 'Design .....	20
3.2 Analyse des besoins et spécification des fonctionnalités .....	20
3.2.1 Identification et représentation des besoins fonctionnels .....	20



3.2.1.1	Services rendus par GASC .....	21
3.2.1.2	Diagrammes des cas d'utilisation .....	21
3.2.2	liste des fonctionnalités du système.....	25
3.2.3	spécification détaillée des besoins.....	25
3.2.3.1	Description du cas d'utilisation « Authentification ».....	25
3.2.3.2	Description du cas d'utilisation « Ajouter dispositif ».....	26
3.2.3.3	Description du cas d'utilisation «Créer topologie ».....	27
3.2.3.4	Description du cas d'utilisation «Générer configurations ».....	27
3.2.3.5	Description du cas d'utilisation « Comparer deux archives ».....	28
3.2.3.6	Description du cas d'utilisation «planifier des tâches ».....	29
3.2.4	Besoins non fonctionnels.....	29
3.3	Spécification du système : modélisation du domaine .....	30
3.4	Conception de la solution .....	31
3.4.1	Architecture logicielle .....	31
3.4.1.1	Identification des composants ou modules du système.....	31
3.4.1.2	Architecture de la solution .....	31
3.4.2	Conception détaillée .....	36
3.4.2.1	Structuration des données.....	36
3.4.2.2	Conception des IHM .....	41
3.4.2.3	Description détaillée des fonctionnalités.....	42
3.5	Conception des algorithmes .....	45
3.5.1	Algorithme de génération des configurations.....	45
3.5.2	Algorithme de sauvegarde des configurations.....	45
3.5.3	Synoptique de l'algorithme de sauvegarde des configurations.....	46
CHAPITRE 3 : MISE EN ŒUVRE ET RESULTATS .....		49
3.1	Choix des outils .....	49
3.1.1	Plateforme de développement : WampServer .....	49

3.1.2	Editeur de code source : Sublime Text 3 .....	49
3.1.3	Système de gestion de base des données : MySQL.....	50
3.2	Mise en place de l'environnement de développement .....	50
3.3	Réalisation des fonctionnalités .....	51
3.3.1	Générateur des configurations .....	51
3.3.2	Sauvegarde des configurations .....	52
3.3.3	Comparateur des fichiers de configurations .....	52
3.4	Déploiement .....	53
3.4.1	Diagramme de déploiement.....	53
3.5	Résultats : quelques scénarii d'utilisation du système .....	55
3.5.1	Processus d'authentification.....	55
3.5.2	L'accueil de l'application .....	56
3.5.3	Processus d'édition d'un nœud .....	56
3.5.4	Processus d'ajout d'une nouvelle interface .....	57
3.6	Validation de la solution : Etude de cas réelle .....	58
3.6.1	Topologie réseau.....	58
3.6.2	Processus d'ajout de topologie.....	59
3.6.3	Création des liens.....	59
3.6.4	Génération des configurations .....	60
3.6.5	Listing des archives des configurations .....	61
3.6.6	Comparaison des fichiers des configurations .....	61
CONCLUSION GENERALE ET PERSPECTIVES.....		63
BIBLIOGRAPHIE .....		65

## LISTE DES FIGURES

Figure 1 : Architecture fonctionnelle.....	8
Figure 2 : Diagramme des cas d'utilisation pour le module génération des configurations .....	22
Figure 3 : Diagrammes des cas d'utilisation pour le module sauvegarde des configurations ....	23
Figure 4 : Diagramme des cas d'utilisation pour le module gestion des utilisateurs .....	24
Figure 5: Diagramme des classes .....	30
Figure 6 : Architecture de l'application GASC.....	32
Figure 7 : Module Générateur des configurations .....	33
Figure 8 : Module sauvegarde des configurations.....	34
Figure 9 : Principe de l'algorithme symétrique [12] .....	35
Figure 10 : Module sécurité des fichiers des configurations .....	36
Figure 11 : Charte graphique de GASC.....	42
Figure 12 : Diagramme de séquence d'authentification.....	42
Figure 13 : Diagramme de séquence Ajouter un dispositif .....	43
Figure 15 : Diagramme de séquence «Comparer deux archives » .....	44
Figure 14 : Diagramme de séquence Créer une topologie réseau .....	44
Figure 16 : Algorithme de génération des configurations .....	45
Figure 17 : Algorithme sauvegarde des configurations.....	46
Figure 18 : Synoptique de l'algorithme de sauvegarde des configurations .....	47
Figure 19 : Wamp server [13].....	49
Figure 20 : Sublime Text 3 [14] .....	49
Figure 21 : Représentation de nœuds .....	53
Figure 22 : Diagramme de déploiement .....	54
Figure 23 : Processus d'authentification .....	55
Figure 24 : L'accueil de l'application.....	56
Figure 25 : Processus d'édition d'un nœud .....	57
Figure 26 : Processus d'ajout d'une nouvelle interface.....	57
Figure 27 : Topologie réseau .....	58
Figure 28 : Processus d'ajout de topologie .....	59
Figure 29 : Création des liens dans GASC .....	59
Figure 30 : Génération des configurations .....	60
Figure 31 : Résultat du ping de PC4 vers le serveur WEB (REUSSITE) .....	60
Figure 32 : Résultat du ping de PC4 vers le serveur WEB (ECHEC).....	60
Figure 33 : Listing des archives des configurations .....	61

Figure 34 : Résultat de la comparaison des fichiers .....	62
---	----

## LISTE DES TABLEAUX

Tableau 1 : Planning de réalisation du travail .....	9
Tableau 3 : Tableau comparatif des solutions .....	16
Tableau 4 : Scenario descriptif de l'authentification .....	25
Tableau 5 : Scenario descriptif Ajouter dispositif .....	26
Tableau 6 : Scenario descriptif Ajouter topologie .....	27
Tableau 7 : Scenario descriptif Générer configurations .....	27
Tableau 8 : Scenario descriptif Comparer deux archives .....	28
Tableau 9 : Scenario descriptif planifier des tâches .....	29
Tableau 10: la table nœuds .....	37
Tableau 11 : table topologie .....	38
Tableau 12 : table utilisateurs .....	38
Tableau 13 : table configurations .....	39
Tableau 14 : table historiques .....	39
Tableau 15 : table taches .....	39
Tableau 16 : table liens .....	40
Tableau 17 : table interfaces .....	40
Tableau 18 : table domaine .....	40
Tableau 19 : table routes_statiques .....	41
Tableau 20: Langages de programmation utilisés .....	49

## INTRODUCTION GENERALE

Le monde du travail actuel tend à migrer vers l'automatisation des tâches par le biais de l'informatisation. Le besoin de facilité, efficacité et fiabilité dans la résolution des problèmes et dans la réalisation des tâches est devenu énorme et pousse les entrepreneurs à doter de leurs entreprises des systèmes de gestion automatisés. De ce fait, pour ce mémoire de fin d'études, nous avons choisi de développer une solution de gestion des configurations pour le Centre des Réseaux IP de Yaoundé en vue d'améliorer son travail au quotidien.

La gestion des réseaux informatiques est toujours un travail pénible, laborieux, sujet aux erreurs et dont la complexité est sans cesse croissante en raison de l'évolution des technologies et du matériel qui entre en compte. D'une part, les équipements qui forment le réseau doivent se comporter comme un groupe. Cependant, chacune de ces machines est gérée et configurée individuellement. La question fondamentale est la même depuis plusieurs années. Un ingénieur réseau est responsable d'un pool de dispositifs dont les configurations individuelles sont gérées pour la plupart manuellement. Chaque fois qu'un nouveau service doit être ajouté aux appareils du réseau, il doit s'assurer du réglage parfait et approprié des paramètres de configuration de ces appareils.

En plus, les sauvegardes des configurations sont effectuées périodiquement par les administrateurs réseaux, en se connectant sur chaque équipement, puis récupérer manuellement les configurations par une simple opération de copier coller dans un bloc-notes, puis d'enregistrer ce dernier sur la machine de l'administrateur réseau. Cette approche, bien qu'elle soit adaptée à un nombre d'équipements réduit, ne l'est pas pour un nombre assez important. Tous les fichiers enregistrés dans les bloc-notes sont souvent mal organisés, ce qui complique la recherche de la dernière sauvegarde. Cette technique entraîne également une mauvaise gestion des archives des configurations. Le fait que les fichiers de configurations soient stockés sur la même machine est un risque non négligeable.

Les sauvegardes sont indispensables pour préserver les données en cas de désastre dans ce monde de plus en plus informatisé où les données représentent les richesses des opérateurs. L'archivage n'est pas non plus à négliger puisqu'il permet de remonter beaucoup plus loin dans le temps afin de faire des enquêtes ou de récupérer des fichiers perdus longtemps dans le passé. Les données ont de la valeur qu'on est prêt à dépenser pour les sauvegarder de manière raisonnée.

Au vu de ces faits, il est clair que de nouveaux paradigmes de configuration sont nécessaires. Dans ce mémoire, nous présenterons une solution applicative qui tente de rencontrer l'ensemble des besoins formulés par les opérateurs. L'application est intitulée *GASC*, l'acronyme de *Gestion Automatisée et Sécurisée des Configurations*. Elle se veut être une solution *flexible* capable de *générer* des configurations réseaux et sauvegarder ces dernières. La flexibilité de l'application est motivée par le fait qu'il est extrêmement difficile de modéliser chaque concept réseau existant ou ayant existé au sein d'une seule et même représentation. Dès lors, nous désirons une application dans lequel l'ajout d'une fonctionnalité soit la plus simple possible. Ainsi, nous permettrons à chaque opérateur de représenter facilement les fonctionnalités dont il a besoin pour configurer son propre réseau. Notre objectif est de rendre l'automatisation des configurations réseaux, la reprise après sinistre, la gestion des modifications et l'audit de configuration simples et abordables.

# CHAPITRE 1 : CONTEXTE ET PROBLEMATIQUE

## 1.1 Contexte du travail

Le cycle d'ingénieur des télécommunications à SUP'PTIC de Yaoundé est clôturé par la présentation d'un mémoire de fin d'études. C'est dans ce cadre que nous sommes tenus de produire le présent travail qui s'est déroulé au sein de l'entreprise CAMTEL.

Plus précisément, ce travail s'est effectué au Centre des Réseaux IP de Yaoundé (CRIPY) où nous avons remarqué que la majorité des équipements réseaux IP sont configurés manuellement ainsi que la sauvegarde des équipements réseaux. Cette approche est sujette aux erreurs de configurations, à la mauvaise gestion des fichiers des configurations, éléments primordiaux dans un réseau informatique.

## 1.2 Problématique

Nous avons constaté au moment de notre arrivée au CRIPY que la majeure partie des configurations réseaux, était faite manuellement. Le constat a été également fait sur la gestion de la sauvegarde des configurations réseaux.

Pour mieux cerner notre problématique, nous allons commencer tout d'abord par expliciter les problèmes liés à la configuration des équipements réseaux, ensuite les problèmes liés à la sauvegarde des configurations des équipements réseaux.

### 1.2.1 Problèmes liés à la configuration des équipements réseaux

Force est de constater que, de nos jours, la configuration manuelle du réseau équipement par équipement via une interface en ligne de commande – ou CLI – reste effectivement le mode opérationnel privilégié. Or, il a été montré à maintes reprises que l'utilisation de ces mécanismes de bas niveau constitue la cause de nombreuses erreurs dans les configurations. En effet, les CLI présentent des inconvénients. En effet, étant donné que les CLI sont des systèmes propriétaires, il est difficile de les utiliser afin d'automatiser des tâches dans un environnement hétérogène au vu des disparités existant entre ces systèmes. Nous posons alors quelques questions concernant la configuration des équipements réseaux :

- ✓ Comment permettre une vérification a priori des configurations afin d'éviter des erreurs;
- ✓ Comment automatiser les configurations réseaux.

### 1.2.2 Problèmes liés à la sauvegarde des configurations réseaux

Un administrateur réseau doit toujours être en mesure de faire face à tous les imprévus. Il doit donc, bien évidemment, s'attendre un jour où l'autre à la panne des équipements réseaux. Lors du remplacement de l'équipement défectueux, il faudra alors remettre la configuration. Pour cela, il aura dû faire une sauvegarde de celle-ci au préalable. Les tâches répétitives doivent être automatisées pour gagner en temps. Parfois les archives de ces configurations sont mal organisées et difficiles à explorer, compliquent ainsi la recherche de la dernière sauvegarde de configurations. Certaines sauvegardes échouent du fait des modifications de la topologie réseau et d'interruptions de services, le fichier de sauvegarde n'étant alors pas créé. On peut se poser les questions suivantes:

- ✓ Comment sauvegarder de façon automatique les configurations des équipements réseaux ?
- ✓ Comment localiser facilement les sauvegardes des configurations ?
- ✓ Comment anticiper de manière fiable les risques de catastrophes ?
- ✓ Comment sécuriser l'accès aux différents fichiers de configurations ?

## 1.3 Objectifs

Dans le but de répondre aux différentes questions suscitées par les problèmes posés plus haut, nous avons défini les objectifs suivants :

### 1.3.1 Objectif général

L'objectif général de ce projet est d'automatiser et sécuriser les configurations des équipements réseaux et leur sauvegarde.

### 1.3.2 Objectifs spécifiques

Le système que nous mettons en place devra permettre de réaliser les objectifs spécifiques ci-après :

- Automatiser les configurations des équipements réseaux ;
- Automatiser la sauvegarde des configurations des équipements réseaux ;
- Réduire le temps et économiser des efforts grâce à des sauvegardes automatisées et planifiées ;



- Récupérer rapidement les données de configuration après une panne d'appareils ou d'échec de configuration ;
- sécuriser l'accès aux fichiers de configurations.

## 1.4 Résultats Attendus

Le système réalisé devra pour cela permettre aux administrateurs réseaux de :

- Générer des configurations valides pour les équipements réseaux ;
- Sauvegarder les configurations des équipements réseaux ;
- Sécuriser les différents fichiers des configurations des équipements ;
- Effectuer des comparaisons des fichiers des configurations.

## 1.5 Méthodologie

Après avoir ressorti le contexte, les objectifs et la problématique de notre projet, nous allons procéder comme suit:

- ✓ Nous étudierons les approches, démarches et outils existants dans le domaine de la gestion des configurations réseaux, pour relever leurs forces et faiblesses respectives ;
- ✓ Voir notamment quels aspects ne sont pas encore traités, ou pas assez convenablement. Ceci nous permettra alors de proposer de nouvelles approches, approfondissant certaines parties de la gestion de configurations réseaux.
- ✓ Nous pourrions dès lors établir des spécifications sur un nouvel outil et définir une architecture, puis le concevoir et le réaliser.
- ✓ Par la suite, nous implémenterons cet outil et l'expérimenterons. Nous observerons alors concrètement son efficacité. Enfin, nous conclurons par quelques recommandations quant à l'évolution de l'outil.

## 1.6 Plan du mémoire

Dans ce qui suit, nous présenterons la structure globale du document. Nous mentionnerons également le contenu de chaque chapitre :

- Le premier chapitre sera consacré au contexte et à la problématique, tout en fixant les objectifs et la méthodologie de travail adoptée, ainsi que la solution proposée après l'étude et les critiques du système existant.
- Le deuxième chapitre fera l'objet de la méthodologie de conception. Dans un premier temps, nous allons faire un état de l'art en matière de gestion des configurations réseaux,

en livrant quelques solutions standardisées par des organismes tels que l'IETF et l'ISO ainsi que la présentation de quelques travaux de recherches universitaires, tout en dégagant les différentes comparaisons entre ces travaux et le nôtre. Dans un second temps, nous allons entamer la conception de la solution en montrant les différents algorithmes utilisés par notre solution. Nous allons également présenter la structure modulaire et l'architecture fonctionnelle de la solution. Nous allons terminer le chapitre par la présentation des différents modèles UML.

- Le troisième et dernier chapitre sera consacré à la réalisation de l'application, en présentant les différents outils et langage de programmations utilisés pour produire l'application. Il se conclura par la présentation des différents scénarios de l'application sous forme de capture d'écran.

## **1.7 Etude de l'existant**

Actuellement à Camtel, il n'existe pas de procédure de vérification a priori des configurations réseaux. La plupart des réseaux IP sont encore configurés manuellement par les administrateurs, routeur par routeur.

Les sauvegardes des configurations sont effectuées périodiquement par les administrateurs réseaux, en se connectant sur chaque équipement et effectuer une opération de copier coller de la configuration courante dans un bloc-notes, puis d'enregistrer ce dernier sur la machine de l'administrateur réseau.

## **1.8 Bilan critique de l'existant**

La démarche utilisée par les administrateurs réseaux pour configurer manuellement les équipements réseaux est sujette aux erreurs et peut être la cause de plusieurs facteurs :

- Interruptions de services ;
- Mauvaise configuration.

Il est donc très difficile d'adopter la configuration manuelle comme approche principale avec le nombre sans cesse croissant des équipements réseaux.

La seconde approche utilisée pour les sauvegardes, bien qu'elle soit adaptée à un nombre d'équipements réduit, ne l'est pas pour un nombre considérable. Tous les fichiers enregistrés dans les bloc-notes sont souvent mal organisés, ce qui complique la recherche de la dernière sauvegarde, entraînant aussi une mauvaise gestion des archives. Ceci est très important

lorsqu'il s'agit de faire des enquêtes ou de récupérer des fichiers perdus longtemps dans le passé. Le fait que les fichiers de configurations soient stockés sur la même machine est un risque non négligeable. Les données de configuration ont de la valeur qu'on est prêt à dépenser pour les sauvegarder de manière raisonnée [1].

Au vu de ces faits, il est clair que de nouveaux paradigmes de configuration sont nécessaires.

## 1.9 Solution Proposée

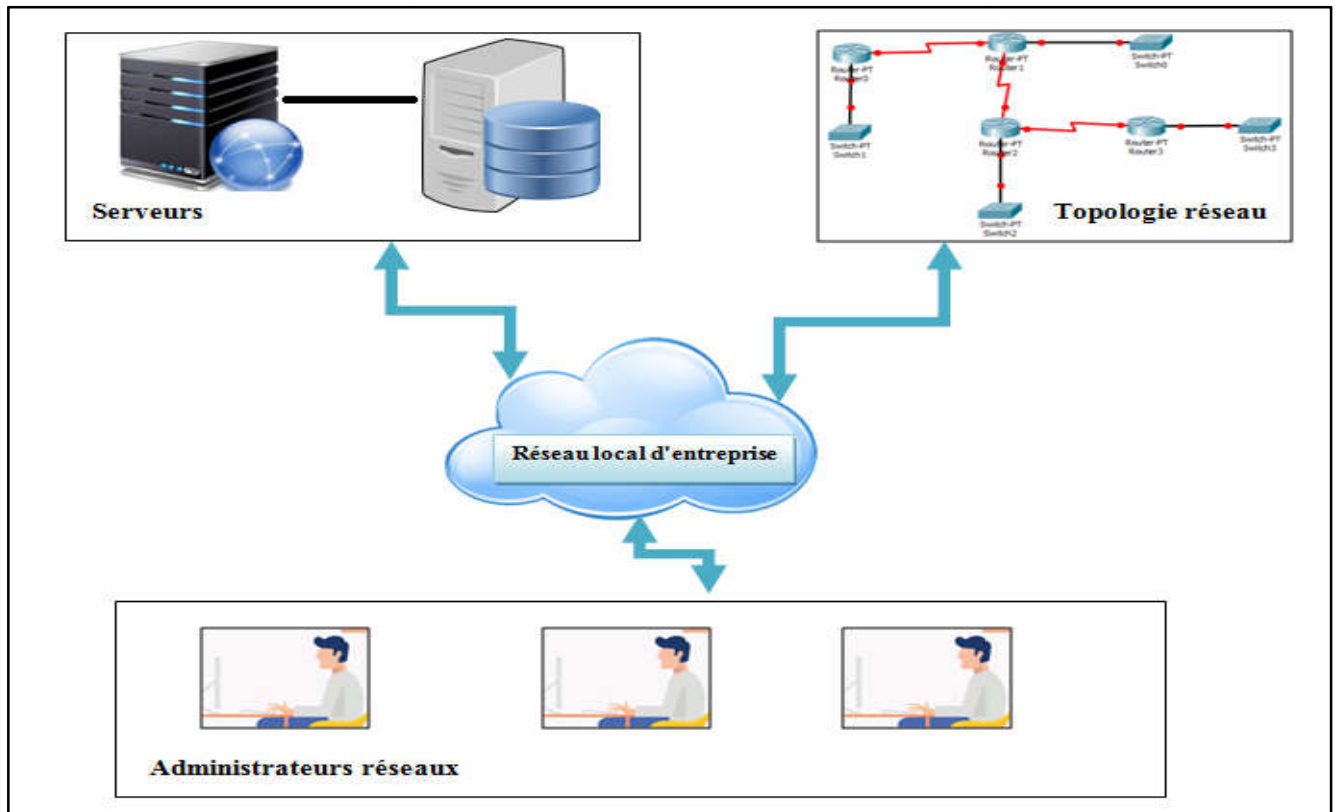
La solution proposée dans ce travail après étude du système actuel et ses limites est une application Web qui fonctionne selon le principe de l'architecture trois-tiers (un tiers pour le client, un tiers pour le serveur d'application et un tiers pour le serveur des données). Dans l'architecture à 3 niveaux (appelée architecture 3-tiers), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre :

- Un client, c'est-à-dire l'ordinateur demandeur de ressources, équipée d'une interface utilisateur (généralement un navigateur web) chargée de la présentation ;
- Le serveur d'application (appelé également middleware), chargé de fournir la ressource mais faisant appel à un autre serveur ;
- Le serveur de base de données, fournissant au serveur d'application les données dont il a besoin.

### Description de l'architecture :

- Les administrateurs réseaux à travers un navigateur internet accèdent à l'application, puis pourront effectuer des opérations de sauvegarde, de génération ou de comparaison d'archives sur la topologie réseau;
- L'architecture comprend un serveur web Apache pour interpréter les scripts PHP et un serveur de base de données pour stocker l'ensemble de données manipulées par GASC ;

La figure 3 présente l'architecture fonctionnelle de notre solution.



**Figure 1 : Architecture fonctionnelle**

### 1.10 Planning de réalisation du projet

La clé principale de la réussite d'un projet est un bon planning. En effet, le planning aide à bien subdiviser le travail et séparer les tâches à réaliser, Il offre une meilleure estimation et gestion de temps nécessaire pour chaque tâche. De plus, il donne assez de visibilité permettant d'estimer approximativement la date d'achèvement de chaque tâche.

Dans ce travail, nous avons estimé la réalisation de notre solution dans une durée approximative de 4,5 mois. Le tableau 1 montre le planning que nous avons adopté pour mener à bien la réalisation des différentes parties du projet.

**Tableau 1 : Planning de réalisation du travail**

<b>Tâches</b>	<b>Date de début</b>	<b>Date de fin</b>
<b>Etude, bilan critique de l'existant</b>	09 / 07 / 2018	13 / 07 / 2018
<b>Problématique</b>	16 / 07 / 2018	18 / 07 / 2018
<b>Solution proposée</b>	19 / 07 / 2018	31 / 07 / 2018
<b>Définition des objectifs</b>	01 / 08 / 2018	06 / 08 / 2018
<b>Résultats attendus</b>	01 / 08 / 2018	06 / 08 / 2018
<b>Revue de la littérature scientifique</b>	07 / 08 / 2018	22 / 08 / 2018
<b>Analyse des besoins et spécification des fonctionnalités</b>	23 / 08 / 2018	31 / 08 / 2018
<b>liste des fonctionnalités du système</b>	23 / 08 / 2018	31 / 08 / 2018
<b>spécification détaillée des besoins</b>	03 / 09 / 2018	07 / 09 / 2018
<b>Spécification du système : modélisation du domaine</b>	10 / 09 / 2018	21 / 09 / 2018
<b>Conception de la solution</b>	24 / 09 / 2018	19 / 10 / 2018
<b>Mise en œuvre de la solution et résultats</b>	22 / 10 / 2018	16 / 11 / 2018
<b>Rédaction du mémoire</b>	19 / 11 / 2018	30 / 11 / 2018

Dans ce chapitre nous avons présenté le contexte du projet, la problématique du projet, les objectifs, la méthodologie, les résultats attendus ainsi que la structure sous forme de chapitre du document. Nous avons également décrit l'état actuel du système de gestion des configurations de la Camtel, relevé ses défauts et proposé une solution.

# CHAPITRE 2 : METHODOLOGIE DE LA CONCEPTION

## I. ETATS DE L'ART

Dans ce chapitre, nous allons tout d'abord présenter un résumé des principaux travaux effectués dans le domaine de la gestion et de la validation des configurations réseaux. Deux grandes parties seront détaillées à savoir quelques solutions standardisées et des travaux de recherche. Étant donné que la dernière partie traite des problématiques similaires à la nôtre, nous tâcherons de dresser une comparaison entre ces travaux et le nôtre. Ensuite nous allons aborder la méthodologie de conception adoptée afin d'aboutir à la solution proposée.

### 2.1 Gestion automatisée des configurations réseaux

L'évolution fulgurante des réseaux informatiques et Internet a fait accroître la charge de travail des administrateurs réseaux, occasionnant ainsi une augmentation des ressources humaines consacrées à leur gestion. Les capacités de gestion de réseau ont été poussées à leurs limites et sont donc devenues plus complexes et source d'erreurs. Dans cette partie, nous allons présenter les enjeux de la configuration manuelle faite équipement par équipement.

#### 2.1.1 Événements de l'actualité

Dans les dernières années, on a assisté à plusieurs incidents liés à des configurations incorrectes des équipements à cause d'une mauvaise vérification des configurations ou tout simplement à cause de la charge importante du travail des administrateurs [2].

Un incident majeur qui a paralysé le réseau Internet pendant une longue période, 20 minutes pour certains et 3 heures pour d'autres, en avril 1997 aux États-Unis. Cet incident est dû suite à une mauvaise configuration des routeurs qui a rendu incorrecte leur table de routage. Cette dernière à entraîner rapidement une inondation des routeurs par le trafic réseau de la société MAI en Virginie, société qui hébergeait les routeurs devant interconnecter les différents réseaux pour la distribution d'Internet.

L'ensemble des erreurs rencontrées ont tous un point commun : **une erreur de configuration**.

#### 2.1.2 Les enjeux

Les enjeux d'avoir une configuration stable et fonctionnelle sont donc énormes quel que soit le nombre de postes ou d'appareils à gérer. Les conséquences d'une mauvaise configuration sont parfois catastrophiques. De nos jours certaines personnes pensent que pour solliciter un système de configuration il faut d'abord avoir un nombre important d'équipements,

ce qui est certes vrai comme première exigence, mais pas comme condition essentielle [3]. Il existe plusieurs causes possibles pour les pannes réseaux, mais les experts s'accordent pour les regrouper en trois catégories à savoir :

- Les événements planifiés de maintenance ;
- Les erreurs système ;
- Les facteurs humains.

Malgré que les vendeurs d'équipements réseaux se focalisent tous sur les deux premières catégories, les erreurs liées à l'action humaine n'en demeurent pas moins la source la plus répandue.

Plusieurs cabinets ont réalisé des études sur les causes et les enjeux des problèmes de configuration et tous sont unanimes que l'intervention humaine en est le principal facteur. Citons d'emblée le cas de la société Juniper Networks qui dans son livre blanc révèle que 50 à 80% des erreurs de configuration réseau sont causées par le facteur humain [4].

Le cas de l'incident de 1997 qui avait paralysé internet, un système de génération de configuration et de vérification automatique aurait pu aider à déceler le changement dans le fichier de configuration du routeur. Dans le passé, toute panne du système, fut-elle minime, nécessitait une intervention humaine, ce qui conduisait à la réduction du temps que pouvaient accorder les administrateurs aux autres tâches.

### 2.1.3 Qu'est-ce qu'une configuration ?

Le terme « gestion de configuration » fait référence à la gestion des configurations techniques des équipements, c'est-à-dire celle qui permet de valider les configurations et de vérifier le bon fonctionnement des équipements au sein d'un réseau [5].

Quand on parle de « configuration » dans le domaine des réseaux informatiques on fait référence à la modification du fichier de configuration, au réglage de paramètres des équipements réseaux comme les routeurs, les pare-feux et les commutateurs intelligents en vue de leur optimisation pour un meilleur fonctionnement. La configuration fait ressortir la notion de fichier et de paramètre. Les fichiers contiennent les paramètres sur lesquels sont basés les configurations afin d'assurer le contrôle et le bon fonctionnement des équipements.

### 2.1.4 Causes

Les configurations sont parfois sujettes à des erreurs qui peuvent occasionner des dysfonctionnements majeurs ou mineurs selon la portée du paramètre ou de la valeur mise en cause. Il peut y avoir plusieurs causes pour ces erreurs dans les fichiers de configuration :

- Erreurs de syntaxe ;
- Erreurs de cohérence.

### 2.1.5 Conséquences

Tout dysfonctionnement dans la configuration d'un équipement peut entraîner de graves conséquences sur la sécurité même de l'infrastructure, du routage des paquets et aussi occasionner des pertes financières énormes.

## 2.2 Approches actuelles dans la gestion des configurations

Dans la section précédente, nous avons présenté plusieurs problèmes ou incidents informatiques liés à une mauvaise configuration d'un équipement réseau dû à l'intervention manuelle de l'homme. L'expérience nous montre que toute intervention manuelle peut être sujette aux erreurs, il serait donc essentiel d'utiliser une machine pour limiter ces incidents.

Dans cette section, nous présenterons un résumé des principaux travaux effectués dans le domaine de la gestion et de la validation de configuration de réseaux. Deux grandes parties seront détaillées : **quelques solutions standardisées ainsi que des travaux de recherche.**

### 2.2.1 Solutions standardisées

#### 2.2.1.1 SNMP [6]

SNMP est un protocole de gestion réseau créé en 1988, du fait de sa simplicité, plusieurs vendeurs l'implémentent. Il a comme objectifs principaux :

- L'automatisation du processus de configuration et de contrôle des éléments du réseau ;
- L'automatisation du processus de supervision des performances du réseau ;
- L'automatisation de la gestion des erreurs au sein du réseau.

Les deux derniers points correspondent à l'automatisation de la tâche de surveillance ou de *monitoring* d'un réseau.

#### ➤ Avantages

En raison de sa standardisation, SNMP a l'avantage d'être interopérable et d'être supporté par la grande majorité des vendeurs.



➤ **Inconvénients**

Cependant, il possède certaines faiblesses inhérentes à la gestion des configurations. Le problème est que l'environnement réseau a fortement changé depuis la conception de SNMP et que les équipements actuels sont capables d'exécuter des opérations beaucoup plus complexes à moindre coût. Pour toutes ces raisons, SNMP est à l'heure actuelle, beaucoup plus utilisé pour des tâches simples de *monitoring* que pour la configuration de réseau.

### 2.2.1.2 NetConf

À cause des problèmes de SNMP évoqués ci-dessus, l'IETF a monté en 2003 un groupe de travail dont l'objectif était d'établir un nouveau protocole de configuration de réseaux. Le résultat de leur travail est le protocole NetConf qui utilise le langage de balisage XML.

➤ **Avantages**

NetConf présente plusieurs avantages. Il a la capacité d'assurer une certaine sécurité de l'échange grâce à l'utilisation de mécanismes d'identification et également de chiffrement lorsque, par exemple, le protocole SSH est utilisé.

➤ **Inconvénients**

NetConf ne correspond pas à notre contexte d'étude, car il ne permet pas la génération des configurations réseaux.

## 2.2.2 Solutions non standardisées

### 2.2.2.1 RAT et Crocodile

Dans le domaine plus spécifique de la sécurité des réseaux, un des outils les plus connus de validation de configuration est nommé RAT. Ce dernier permet de valider des configurations basées sur le langage IOS de Cisco Systems. L'outil Crocodile suit le même principe que RAT et permet, lui-aussi, de valider des configurations de routeurs, toujours d'un point de vue sécurité.

### Comparaison

La philosophie suivie par ces deux outils diffère de notre approche dans le sens qu'ils valident des configurations existantes. Notre solution permet de générer des configurations validées. Un autre point important est le fait que RAT et Crocodile se concentrent sur l'aspect sécurité des configuration, là où notre travail n'est pas restreint à un domaine particulier.

#### 2.2.2.2 NetML

Une autre approche intéressante est celle adoptée par NetML. Il s'agit d'un langage basé sur XML.

##### ➤ **Avantages**

Il Permet de décrire des réseaux à un haut niveau. L'outil basé sur NetML est capable de générer des configurations *concrètes* supportées par différents vendeurs tels que Cisco Systems, Juniper.

##### ➤ **Inconvénients**

NetML ne valide pas la configuration représentée et ne permet pas de gérer les sauvegardes.

#### 2.2.2.3 MetaConfig

Le travail proposé par **Miroslav Matuska** dans *Metaconfiguration of the computer network (metaconfig)* [7] part du même constat que NetML: L'auteur propose une représentation de haut niveau du réseau.

##### ➤ **Avantages**

Il est possible, à partir de la représentation de haut niveau, de générer les configurations propres à chaque constructeur.

Ce travail nous a beaucoup inspiré, du fait qu'il traite un des aspects de notre solution, à savoir *la génération des configurations*.

##### ➤ **Inconvénients**

Néanmoins, il n'aborde pas la problématique de la sauvegarde des configurations réseaux.

#### 2.2.2.4 El-Arini et Killourhy

El-Arini et Killourhy [8] proposent un algorithme de détection d'anomalies au sein de configurations.

##### ➤ **Avantages**

Cet algorithme se base sur des principes statistiques et plus spécialement sur un détecteur bayésien. Les auteurs l'ont testé sur un réseau universitaire présentant des erreurs. L'outil a permis de découvrir la plupart de ces erreurs avec un taux de faux positifs raisonnable.

### ➤ Inconvénients

Cet algorithme est beaucoup plus orienté pour découvrir les erreurs des configurations. Il ne permet pas de générer ni de sauvegarder les configurations.

#### 2.2.2.5 VNG

Le travail proposé par L. VANBEVER et G. PARDOEN [9] aboutit à un outil qu'ils ont nommé VNG.

### ➤ Avantages

Il se veut être une solution *flexible* capable de *générer* des configurations de réseaux *validées*. VNG, se base essentiellement sur deux grands principes. En premier lieu, il utilise une représentation de haut niveau afin de modéliser l'entièreté d'un réseau sous une seule entité. Cette dernière est décrite sous la forme d'un fichier XML. Une telle représentation permet, entre autres, d'éviter la redondance et d'être totalement indépendante d'un vendeur particulier. En second lieu, il se base sur un ensemble de règles qui valident cette représentation.

### ➤ Inconvénients

Le problème avec leur approche, est qu'un opérateur doit assurer lui même le procédé de traduction de son réseau au sein de leur représentation, ce qui s'avère être une perte de temps et sujette aux erreurs. En plus la problématique de la sauvegarde des configurations n'est pas prise en compte dans leur démarche.

## 2.3 Etude comparative des solutions

Dans cette partie, nous allons décrire brièvement une étude comparative des solutions étudiées plus haut en fonction des critères que nous avons jugés utiles dans le contexte de la gestion des configurations.

### 2.3.1 Définition des critères de comparaison

Pour effectuer une comparaison objective, les critères définis sont les suivants :

- **Disponibilité de la documentation** : La documentation logicielle est un texte écrit qui accompagne le **logiciel** informatique. Elle explique **comment** le **logiciel** fonctionne et comment on doit l'employer.
- **Sécurité des fichiers des configurations** : c'est un critère très important quand on parle des données très sensibles et qui se trouvent dans un environnement hostile (le réseau).

- **Facilité d'utilisation** : la solution doit être facile d'utilisation, afin de ne pas perdre l'utilisateur quand à l'exécution des fonctionnalités
- **Modularité** : la modularité est un **élément déterminant** dans le choix d'un logiciel, la solution doit être modulaire, afin d'être flexible.
- **Administration** : la gestion des profils utilisateurs, est un critère très important lorsque la solution est multi utilisateurs.

La grille d'évaluation retenue est la suivante :

Nous mettons **1** devant une solution qui remplit le critère et **0** dans le cas contraire. Enfin, une note est donnée pour l'évaluation finale (somme des notes).

### 2.3.2 Tableau comparatif

**Tableau 2 : Tableau comparatif des solutions**

Solutions	Critères	Notes	Evaluation finale
VNG	disponibilité de la documentation	1	3/5
	sécurité des fichiers	0	
	facilité d'utilisation	1	
	Modularité	1	
	Administration	0	
El-Arini et Killourhy	disponibilité de la documentation	0	2/5
	sécurité des fichiers	0	
	facilité d'utilisation	1	
	Modularité	1	
	Administration	0	
MetaConfig	disponibilité de la documentation	1	2/5
	sécurité des fichiers	0	
	facilité d'utilisation	1	
	Modularité	0	

	<b>Administration</b>	<b>0</b>	
<b>NetML</b>	<b>disponibilité de la documentation</b>	<b>1</b>	<b>2/5</b>
	<b>sécurité des fichiers</b>	<b>0</b>	
	<b>facilité d'utilisation</b>	<b>1</b>	
	<b>Modularité</b>	<b>0</b>	
	<b>Administration</b>	<b>0</b>	
<b>RAT et Crocodile</b>	<b>disponibilité de la documentation</b>	<b>1</b>	<b>4/5</b>
	<b>sécurité des fichiers</b>	<b>1</b>	
	<b>facilité d'utilisation</b>	<b>1</b>	
	<b>Modularité</b>	<b>0</b>	
	<b>Administration</b>	<b>1</b>	
<b>GASC</b>	<b>disponibilité de la documentation</b>	<b>1</b>	<b>5/5</b>
	<b>sécurité des fichiers</b>	<b>1</b>	
	<b>facilité d'utilisation</b>	<b>1</b>	
	<b>Modularité</b>	<b>1</b>	
	<b>Administration</b>	<b>1</b>	

Ce tableau comparatif montre que notre solution GASC, respecte tous les critères objectifs établis.

Les domaines de recherche dans la gestion et la validation de configuration de réseaux sont extrêmement féconds au vu du nombre de publications scientifiques et des solutions proposées. Cela est notamment du au fait que la communauté scientifique prend conscience de l'utilité de telles approches. Nous avons présenté deux solutions standardisées dans le domaine plus général de la gestion de configuration réseaux : **SNMP** et **NetConf**. Ces approches bien que très évoluées diffèrent de la nôtre, dans le sens où elles ne permettent pas de valider *a priori* la configuration d'un réseau. Ensuite, nous avons étudié quelques solutions de recherches universitaires et ressorti les comparaisons entre les approches utilisées dans ces travaux et notre

solution. La partie suivante abordera la méthodologie de conception adoptée à fin d'aboutir à notre solution proposée.

## II. CONCEPTION DE LA SOLUTION PROPOSEE

Dans cette partie nous conceptualisons la solution proposée. Pour cela, ce chapitre présente tout d'abord la méthode et le langage d'analyse utilisés, ensuite détaille les besoins spécifiques et en fin la présente quelques diagrammes UML.

### 3.1 Méthodes et langage d'analyse

Comme n'importe quel type de projet, un projet informatique nécessite une phase d'analyse, suivie d'une étape de conception. Dans **la phase d'analyse**, on cherche d'abord à bien comprendre et à décrire de façon précise les besoins des utilisateurs, C'est ce qu'on appelle « **l'analyse des besoins** ». Après validation de notre compréhension du besoin, nous imaginons la solution. C'est la partie **analyse de la solution**. [10]

Dans **la phase de conception**, on apporte plus de détails à la solution et on cherche à clarifier des aspects techniques, tels que l'installation des différentes parties logicielles à installer sur du matériel.

Pour réaliser ces deux phases dans un projet informatique, nous utilisons des méthodes, des conventions et des notations. Pour cela, MERISE et UML sont deux grands principes de « traduction » ou modélisation d'un système d'information. Néanmoins, ils ne sont pas aussi proches qu'on pourrait le penser. Une étude comparative entre MERISE et UML est donnée dans *l'annexe A*.

Notre choix s'est porté sur UML, car c'est un langage le plus utilisé actuellement et est surtout adapté pour des applications ayant une démarche orientée objet, ce qui est le cas dans notre solution.

#### 3.1.1 Langage de modélisation UML

La modélisation consiste à créer une représentation simplifiée d'un problème: **le modèle**. Grâce au modèle il est possible de représenter simplement un problème, un concept et le simuler. La modélisation comporte deux composantes :

- L'analyse, c'est-à-dire l'étude du problème ;

- la conception, soit la mise au point d'une solution au problème Le modèle constitue ainsi une représentation possible du système pour un point de vue donné.

Le langage de modélisation UML respecte un certain nombre de règles sur les concepts manipulés (classes, attributs, opérations, paquetages...) ainsi que sur la syntaxe d'écriture et le formalisme de représentation graphique. L'ensemble de ces règles constitue en soi un langage de modélisation qui a fait l'objet d'un **méta modèle** UML. L'intérêt de disposer d'un méta-modèle UML permet de bien maîtriser la structure d'UML et de faciliter son évolution. [11]

UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux vues.

Une vue est constituée d'un ou plusieurs diagrammes. On distingue deux types de vues :

❖ **la vue statique permettant de représenter le système physiquement :**

- diagrammes de classes : représentant des collections d'éléments de modélisation statique (classes, paquetages) qui montrent la structure d'un modèle ;
- diagrammes d'objets : ces diagrammes montrent des objets (instances d'une classe) et des liens (relations sémantiques) entre objets ;
- diagrammes de cas d'utilisation : identifie les acteurs du système et leurs interactions avec ce dernier ;
- diagrammes de composants : permettent de décrire l'architecture physique statique d'une application en termes de modules : fichiers sources, bibliothèques exécutables ;
- diagrammes de déploiement : montre la disposition physique du matériel qui compose le système et la répartition des composants sur ce matériel ;

❖ **la vue dynamique, montrant le fonctionnement du système :**

- diagrammes de collaboration : montrent des interactions entre objets ;
- diagrammes séquence : permettent de représenter des collaborations entre objets selon un point de vue temporel, on y met l'accent sur la chronologie (envoi des messages) ;
- diagrammes d'état-transitions : permettent de décrire les changements d'états d'un objet ;
- diagrammes d'activités : servent à représenter graphiquement le comportement d'une méthode ou déroulement d'un cas d'utilisation.

Ces diagrammes, d'une utilité variable selon les cas, ne sont pas nécessairement tous produits à chaque modélisation. La conception de notre application a été élaborée suivant la démarche suivante :

- élaboration du diagramme de cas d'utilisation. Cette étape a été réalisée suite à la spécification fonctionnelle de l'application ;
- élaboration des diagrammes d'activités et de séquences pour mettre en évidence les interactions entre les différents objets du système.

En conclusion, nous avons choisi de travailler avec UML parce qu'il exprime mieux la vue statique et dynamique du système d'information et pour notre application web, il est nécessaire de faire une analyse très approfondie pour pouvoir dégager les nécessités de développement ainsi que quelques scénarios d'exécution.

### 3.1.2 Win 'Design

De nombreux outils de modélisation facilitent la structuration des données. Selon l'outil choisi, on peut :

- Concevoir des modèles selon le formalisme UML, MERISE ;
- Générer automatiquement le script pour réaliser la structure de la base de données sous différents SGBD (Oracle, Access, MySQL...) ;
- Générer automatiquement le dictionnaire.

Notre choix s'est porté sur : **Win 'Design**. Il permet de modéliser les traitements informatiques et leurs bases de données. Nous avons utilisés cet outil pour modéliser les diagrammes des cas d'utilisation, le diagramme des classes et les diagrammes de séquences.

## 3.2 Analyse des besoins et spécification des fonctionnalités

### 3.2.1 Identification et représentation des besoins fonctionnels

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données.

Dans cette solution, nous avons la présence d'un seul acteur qui est :

**L'Administrateur du réseau** : considéré comme l'acteur principal de notre application, il bénéficie de toutes les fonctionnalités offertes par l'application ;



### 3.2.1.1 Services rendus par GASC

Après analyse des besoins, nous avons recensé les différents services qu'offrent GASC à l'administrateur :

- Génération des configurations ;
- Sauvegarde des configurations ;
- Sécurisation des configurations ;
- Gestion des utilisateurs du système.

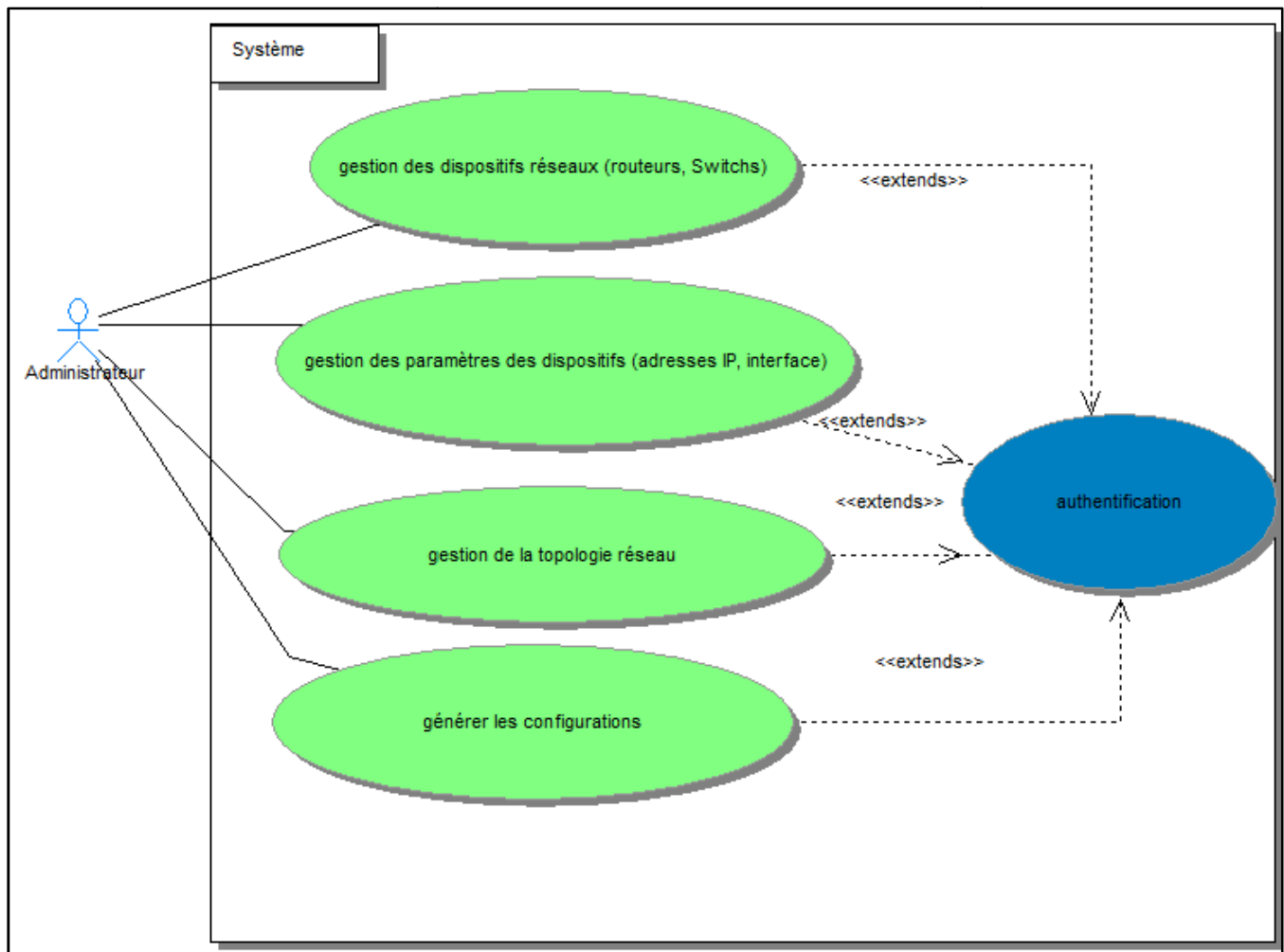
### 3.2.1.2 Diagrammes des cas d'utilisation

#### ➤ Module : Génération des configurations réseaux

Ce module est constitué des différentes fonctionnalités ci après :

- **gestion des dispositifs réseaux** : c'est une fonctionnalité qui permet aux acteurs de gérer les différents dispositifs réseaux. Dans notre cas, ce sont les routeurs et les Switchs. Il s'agit précisément d'effectuer les opérations telles que : **l'ajout, la modification et la suppression** de ces dispositifs dans notre base de données;
- **gestion des paramètres des dispositifs** : ce cas permet aux acteurs de mettre à jour les différents paramètres (**adresses IP, interfaces**) des dispositifs;
- **gestion de la topologie réseau** : c'est un cas qui permet aux acteurs de gérer efficacement la topologie réseau, notamment en **créant la topologie avec ces différents éléments tels que : les dispositifs, les liens**;
- **générer les configurations** : c'est le dernier cas qui intervient après avoir créé la topologie, et permet de générer les différentes configurations de tous les dispositifs de la topologie.

Le diagramme de cas d'utilisation ci-après met en exergue les supers cas d'utilisations pour notre module de la **génération des configurations**.



**Figure 2 : Diagramme des cas d'utilisation pour le module génération des configurations**

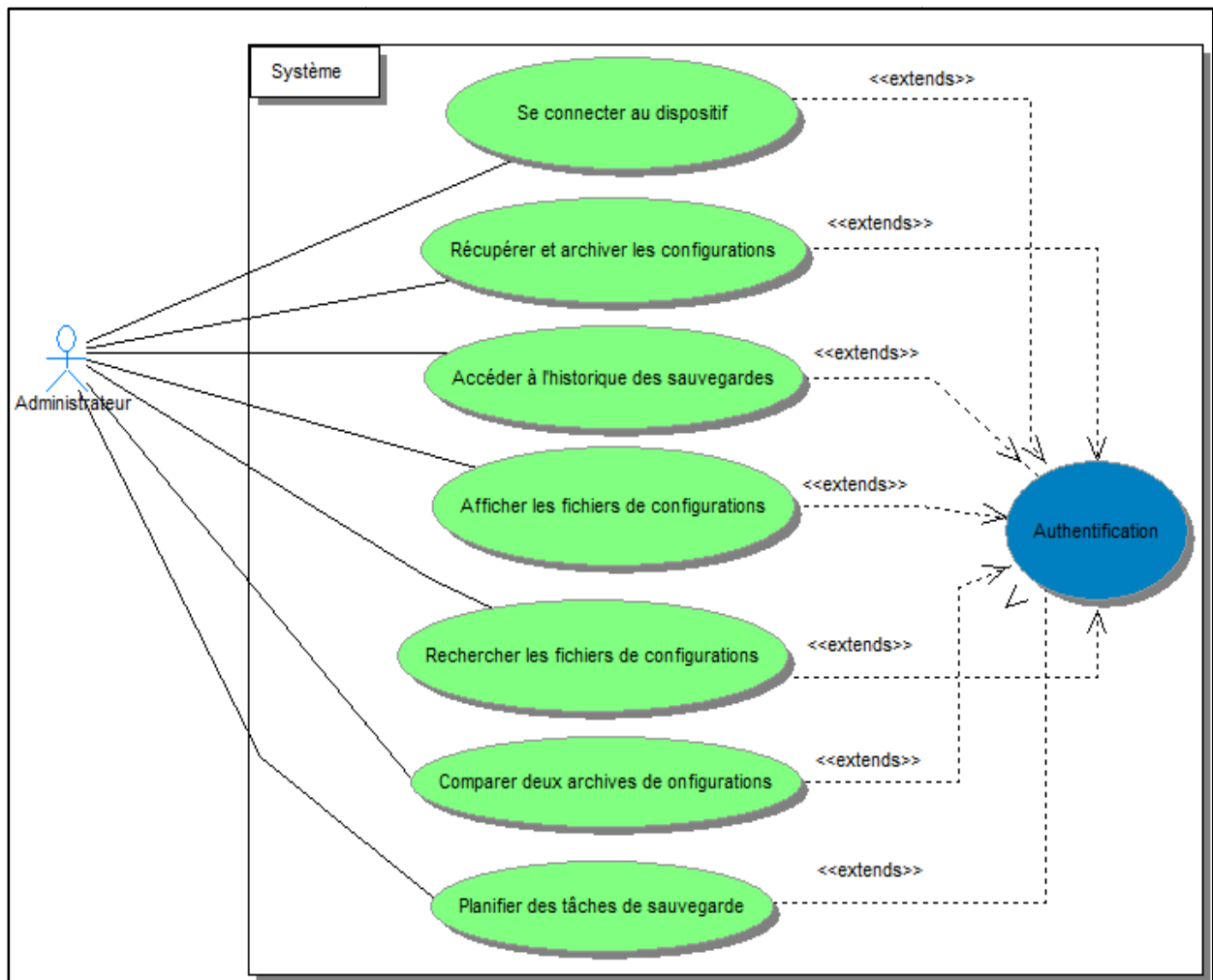
### ➤ **Module : Sauvegarde des configurations**

Ce module est constitué des différentes fonctionnalités ci après :

- **Se connecter au dispositif** : c'est une fonctionnalité permettant d'établir une connexion (Telnet ou SSH) sur l'ensemble des dispositifs ;
- **Récupérer et archiver les configurations** : ce cas permet aux acteurs, après connexion à un dispositif de rapatrier la configuration de ce dernier et de la stocker dans notre base de données ;
- **accéder à l'historique des sauvegardes** ;
- **afficher les fichiers de configurations** ;
- **rechercher les fichiers de configurations** ;

- **comparer deux archives de configurations** : permet d'identifier rapidement où des modifications spécifiques ont été apportées. Cela permet d'annuler rapidement et en toute confiance les modifications indésirables.;
- **planifier des tâches de sauvegarde** ;

Le diagramme de cas d'utilisation ci-après met en exergue les supers cas d'utilisations pour notre module de la **sauvegarde des configurations**.



**Figure 3 : Diagrammes des cas d'utilisation pour le module sauvegarde des configurations**

➤ **Module : gestion des utilisateurs.**

ce module est constitué des différentes fonctionnalités ci-après :

- **gestion des utilisateurs** : c'est un cas permettant à l'administrateur de créer, modifier ou supprimer les utilisateurs de l'application ;
- **consulter l'historique des actions des utilisateurs** ;
- **paramétrer l'application** : c'est un cas permettant à l'administrateur de définir le réglage pour l'application, comme suit :
  - activer ou désactiver le cryptage des identifiants de connexion au dispositif en utilisant l'algorithme de chiffrement symétrique **AES-256-CBC**;
  - Mettre à jour les identifiants de connexion (nom d'utilisateur, mot de passe) aux dispositifs ;
  - changer le mot de passe d'accès à l'application ;

Le diagramme de cas d'utilisation ci-après met en exergue les supers cas d'utilisations pour notre module de la **gestion des utilisateurs**.

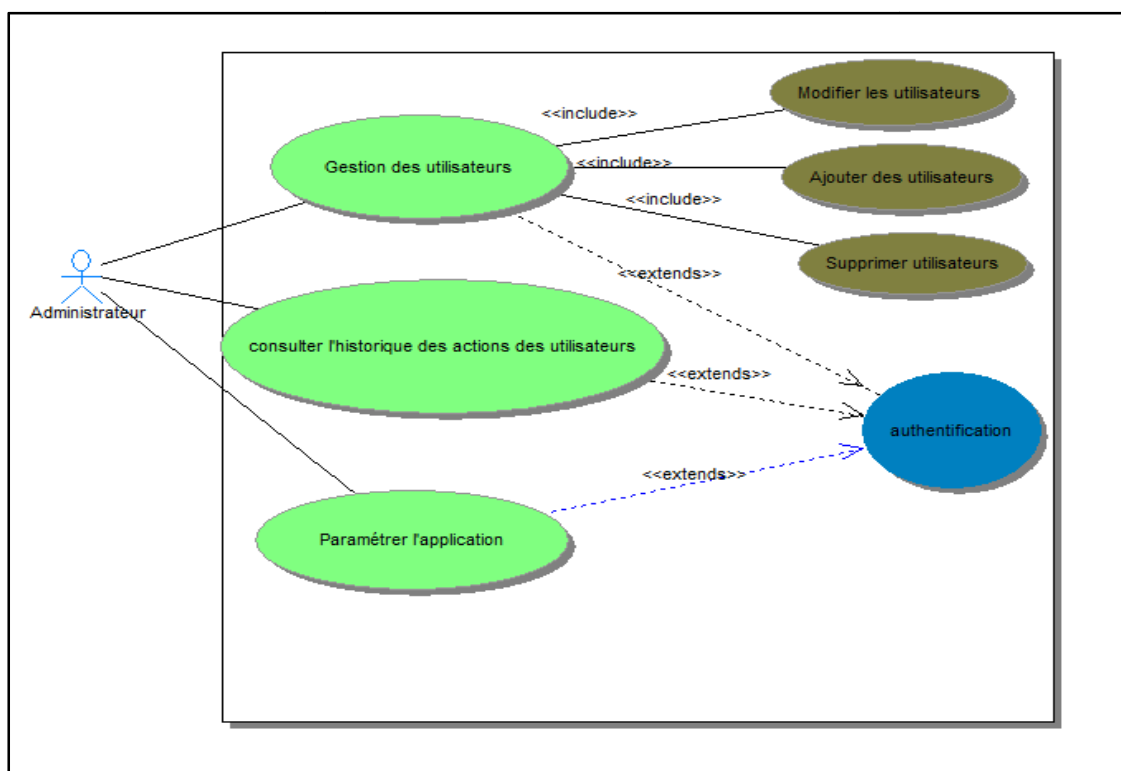


Figure 4 : Diagramme des cas d'utilisation pour le module gestion des utilisateurs

### 3.2.2 liste des fonctionnalités du système

L'analyse du sujet nous a permis de dégager les fonctionnalités qui seront mise à la disposition de l'utilisateur. Les besoins fonctionnels servent à présenter les actions que doit effectuer le système en réponse à une demande présentée par un utilisateur. Dans cette partie, nous allons recenser les fonctionnalités que l'outil doit offrir.

L'application doit réaliser les fonctionnalités suivantes :

- Se connecter via SSH ou TELNET aux dispositifs réseaux (routeurs, Switchs) ;
- Accéder à l'historique des sauvegardes ;
- Récupérer et archiver les configurations ;
- Afficher les fichiers de configurations ;
- Rechercher les fichiers de configurations ;
- Planifier des tâches de sauvegarde ;
- Comparer deux archives de configurations ;
- Gestion des dispositifs réseaux (routeurs, Switchs) ;
- Gestion des paramètres des dispositifs (adresses IP, interfaces, etc.) ;
- Gestion de la topologie réseau ;
- Générer les configurations ;
- Authentification des utilisateurs ;
- Gestion du profil des utilisateurs ;
- Consulter l'historique des actions des utilisateurs ;

### 3.2.3 spécification détaillée des besoins

#### 3.2.3.1 Description du cas d'utilisation « Authentification »

**Tableau 3 : Scenario descriptif de l'authentification**

<b>Nom :</b> Authentification
<b>Acteurs :</b> Tous
<b>Pré-condition :</b> l'acteur possède un profil

**Scenario nominal :**

1 : le système invite l'acteur à entrer son login et son mot de passe.

2 : l'acteur saisit les informations demandées par le système.

3 : le système vérifie les informations saisies.

4 : le système ouvre l'espace de travail correspondant au profil.

**Scenario alternatif :**

Le login ou le mot de passe est incorrect : ce scenario commence au point 3 du scénario nominal.

3. a : le système informe l'acteur que les informations fournies sont erronées et le scenario reprend au point 1 du scenario nominal.

### 3.2.3.2 Description du cas d'utilisation « Ajouter dispositif »

**Tableau 4 : Scenario descriptif Ajouter dispositif**

Nom : Ajouter dispositif	
Acteurs : Administrateur	
Pré-condition : l'acteur est authentifié	
<b>Scenario nominal :</b>  1 : le système affiche un formulaire d'ajout à l'acteur. 2 : l'acteur remplit le formulaire de création de dispositif puis valide. 3 : le système vérifie la validité des informations saisies. 4 : le système enregistre ces informations dans la base de données. 5 : le système notifie l'acteur du bon déroulement de l'opération.	
<b>Scenario alternatif :</b>  Les informations sont manquantes ou incorrectes : ce scenario commence au point 2 du scenario nominal.	

3. a : le système informe l'acteur que les informations fournies sont erronées ou elles sont déjà présentes et le scenario reprend au point 1 du scenario nominal.

### 3.2.3.3 Description du cas d'utilisation «Créer topologie »

Tableau 5 : Scenario descriptif Ajouter topologie

<b>Nom :</b> Créer topologie
<b>Acteurs :</b> Administrateur
<b>Pré-condition :</b> l'acteur est authentifié
<b>Scenario nominal :</b>  1 : le système affiche un formulaire de création de topologie à l'acteur. 2 : l'acteur ajoute au moins deux dispositifs pour la topologie. 3 : l'acteur définit les liens entre les dispositifs. 3 : le système vérifie la validité des informations saisies. 4 : le système enregistre ces informations dans la base de données. 5 : le système notifie l'acteur du bon déroulement de l'opération.
<b>Scenario alternatif :</b>  Les informations sont manquantes ou incorrectes : ce scenario commence au point 2 du scenario nominal. 3. a : le système informe l'acteur que les informations fournies sont erronées ou elles sont déjà présentes et le scenario reprend au point 1 du scenario nominal.

### 3.2.3.4 Description du cas d'utilisation «Générer configurations »

Tableau 6 : Scenario descriptif Générer configurations

<b>Nom :</b> Générer configurations
<b>Acteurs :</b> Administrateur

<b>Pré-condition :</b> l'acteur est authentifié, topologie créée et valide.
<b>Scenario nominal :</b>  1 : le système affiche un formulaire de création de topologie à l'acteur. 2 : l'acteur ajoute au moins deux dispositifs pour la topologie. 3 : l'acteur définit les liens entre les dispositifs. 4 : le système vérifie la validité des informations saisies. 5 : le système enregistre ces informations dans la base de données. 6 : le système notifie l'acteur du bon déroulement de l'opération.
<b>Scenario alternatif :</b>  Les informations sont manquantes ou incorrectes : ce scénario commence au point 2 du scénario nominal.  3. a : le système informe l'acteur que les informations fournies sont erronées ou elles sont déjà présentes et le scénario reprend au point 1 du scénario nominal.

### 3.2.3.5 Description du cas d'utilisation « Comparer deux archives »

Tableau 7 : Scenario descriptif Comparer deux archives

<b>Nom :</b> Comparer deux archives
<b>Acteurs :</b> Administrateur
<b>Pré-condition :</b> l'acteur est authentifié, existence de deux archives de configurations.
<b>Scenario nominal :</b>  1 : le système affiche un formulaire de Comparaison des archives. 2 : l'acteur sélectionne les dispositifs avec leurs archives respectives. 3 : le système vérifie la validité des informations sélectionnées. 4 : le système affiche le résultat de la comparaison.
<b>Scenario alternatif :</b>  Les informations sont manquantes ou incorrectes : ce scénario commence au point 2 du scénario nominal.  3. a : le système informe l'acteur que les informations fournies sont erronées et le scénario



reprend au point 1 du scenario nominal.

### 3.2.3.6 Description du cas d'utilisation «planifier des tâches »

**Tableau 8 : Scenario descriptif planifier des tâches**

<b>Nom :</b> Planifier des tâches
<b>Acteurs :</b> Administrateur
<b>Pré-condition :</b> l'acteur est authentifié
<b>Scenario nominal :</b>  1 : le système affiche un formulaire de Planification des tâches à l'acteur. 2 : l'acteur remplit les paramètres de la tâche. 3 : le système vérifie la validité des informations saisies. 4 : le système enregistre ces informations dans la base de données. 5 : le système notifie l'acteur du bon déroulement de l'opération.
<b>Scenario alternatif :</b> Les informations sont manquantes ou incorrectes : ce scenario commence au point 2 du scenario nominal. 3. a : le système informe l'acteur que les informations fournies sont erronées ou elles sont déjà présentes et le scenario reprend au point 1 du scenario nominal.

### 3.2.4 Besoins non fonctionnels

Les besoins non fonctionnels sont importants car ils agissent de façon indirecte sur le résultat et sur le rendement de l'utilisateur. Ainsi, ils ne doivent pas être négligés. Il faut donc répondre aux exigences suivantes :

- **Fiabilité** : l'application doit fonctionner de façon cohérente sans erreurs ;
- Les erreurs : l'application doit les signaler par des messages d'erreurs ;
- **Ergonomie et bon ihm** : l'application doit être adaptée à l'utilisateur sans qu'il fournisse trop d'effort (utilisation claire et facile) ;
- **Efficacité** : l'application doit permettre l'accomplissement de la tâche avec le minimum de manipulations ;

- **Sécurité** : sécuriser les données revient à appliquer une stratégie d'identification, d'authentification, l'autorisation et contrôler chaque tentative d'accès à ces données.

### 3.3 Spécification du système : modélisation du domaine

Le diagramme de classes exprime la structure statique du système en termes de classes et de relations entre ces classes. L'intérêt du diagramme de classes est de modéliser les entités du système d'information. Ces informations sont regroupées ensuite dans des classes.

La figure suivante présente les classes et leur relation :

Le diagramme de classe de notre solution est basé sur les règles de gestion suivantes :

- Un domaine peut contenir une ou plusieurs topologies réseaux ;
- La topologie réseau est un ensemble de dispositifs et de liens ;
- Un dispositif peut avoir une ou plusieurs interfaces ;
- Un lien est situé entre deux interfaces de dispositifs différents ;
- Un dispositif peut avoir une ou plusieurs configurations.
- un utilisateur peut planifier une ou plusieurs tâches de sauvegarde.

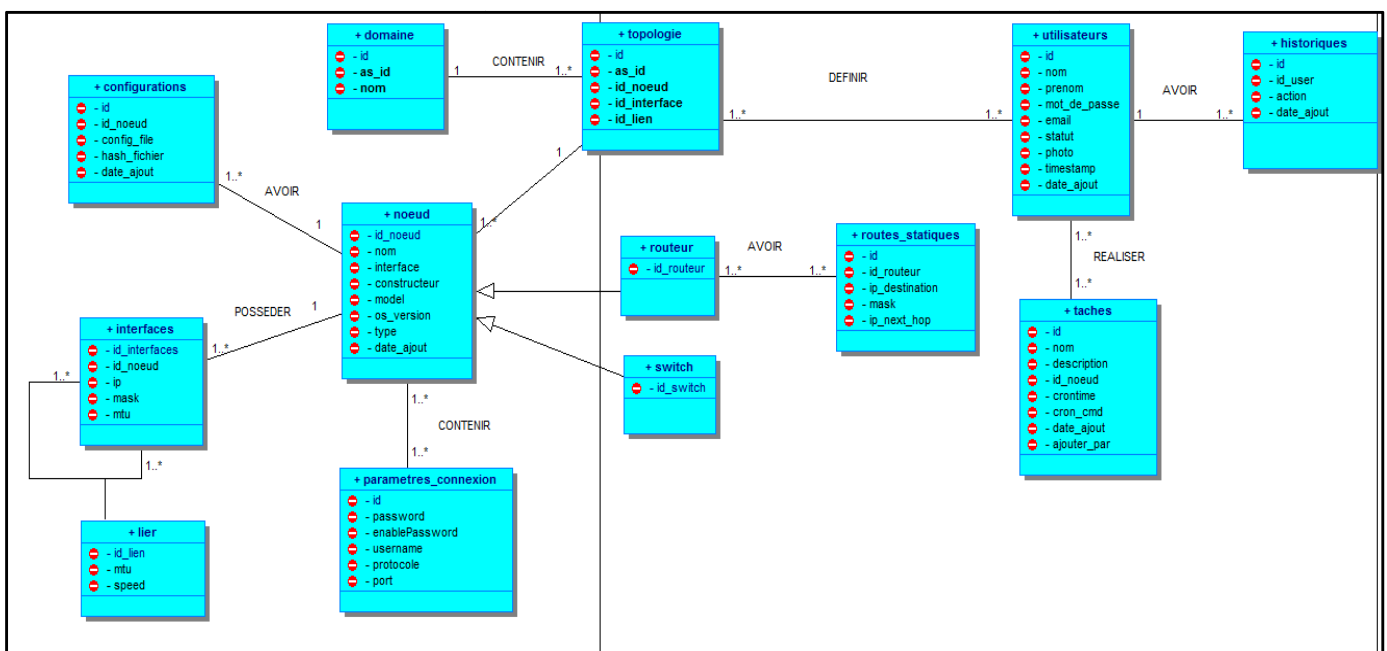


Figure 5: Diagramme des classes

## 3.4 Conception de la solution

### 3.4.1 Architecture logicielle

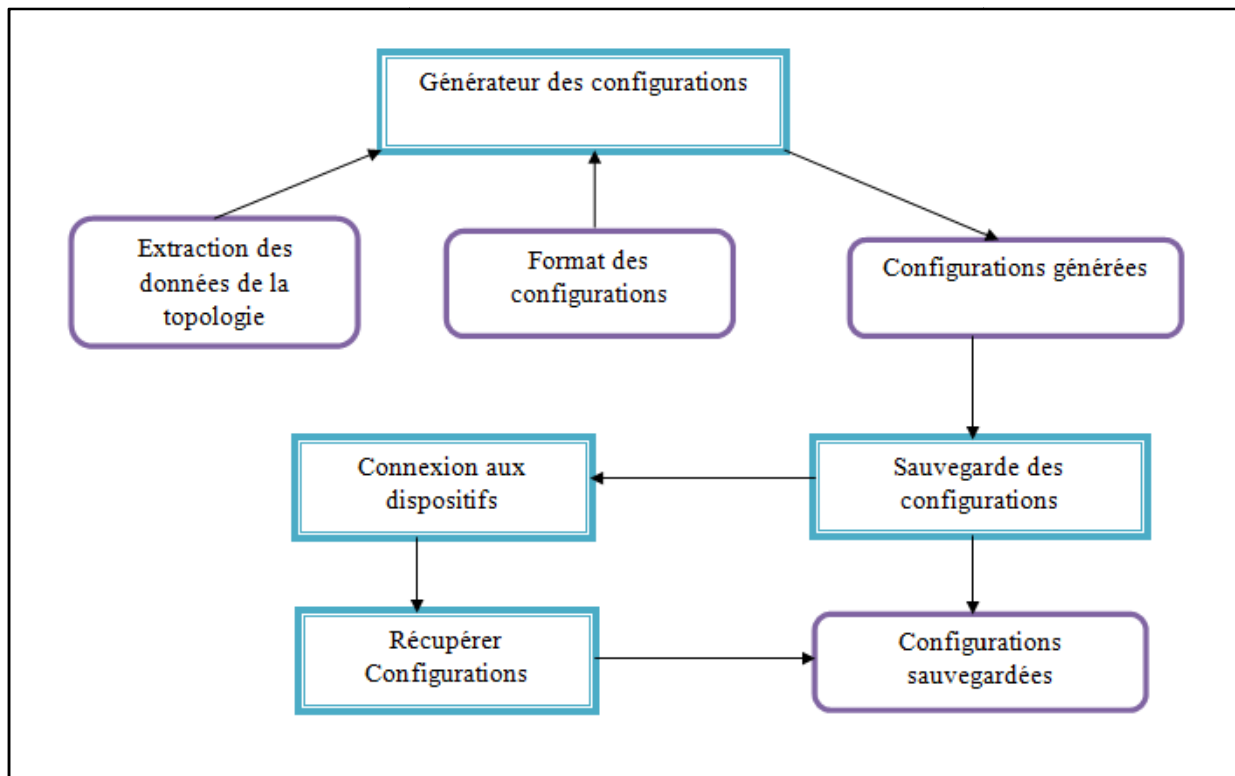
#### 3.4.1.1 Identification des composants ou modules du système

L'application Web GASC à mettre en place est constituée des modules suivants :

- **Module générateur des configurations** : ce composant se charge de produire les configurations des équipements réseaux ;
- **Module sauvegarde des configurations** : Ce module est chargé de rapatrier les configurations des équipements dans un serveur de base de données ;
- **Module sécurité des fichiers des configurations** : ce module regroupe les fonctionnalités qui permettent de chiffrer les fichiers des configurations, de contrôler l'accès à ces fichiers ainsi que la vérification de l'intégrité de ces fichiers ;
- **Module gestion des utilisateurs** : ce module est chargé de gérer les différents utilisateurs de GASC. Il regroupe des fonctionnalités qui permettent d'ajouter, de modifier ou de supprimer un utilisateur.

#### 3.4.1.2 Architecture de la solution

Afin de réaliser tous les objectifs fixés, notre application doit être structurée afin d'être un maximum flexible. La figure suivante montre son architecture.



**Figure 6 : Architecture de l'application GASC**

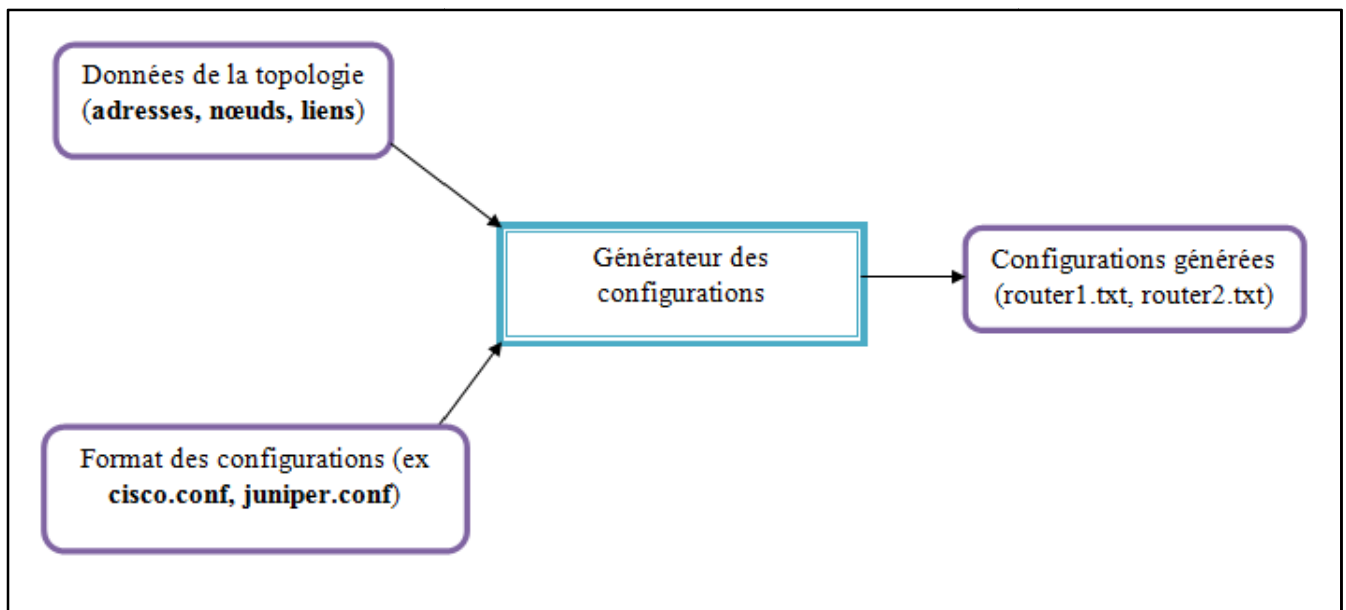
*GASC* est composée de deux processus principaux : **le générateur et la sauvegarde des configurations**.

➤ **Générateur des configurations**

Le générateur permet de produire les configurations souhaitées des différents dispositifs de la topologie réseau. Ce processus a besoin de deux entrées : le format des configurations (c'est un fichier qui décrit le langage des configurations) et les données de la topologie réseau (adresses IP, nœuds, liens entre les nœuds).

Dès que les formats des configurations et la topologie réseau valide sont disponibles, les configurations réelles destinées à être déployées sur les équipements du réseau peuvent être générées. Dans cette section, nous expliquerons comment ceci est réalisé par notre application.

La figure suivante montre la structure du *générateur* permettant de produire les configurations.

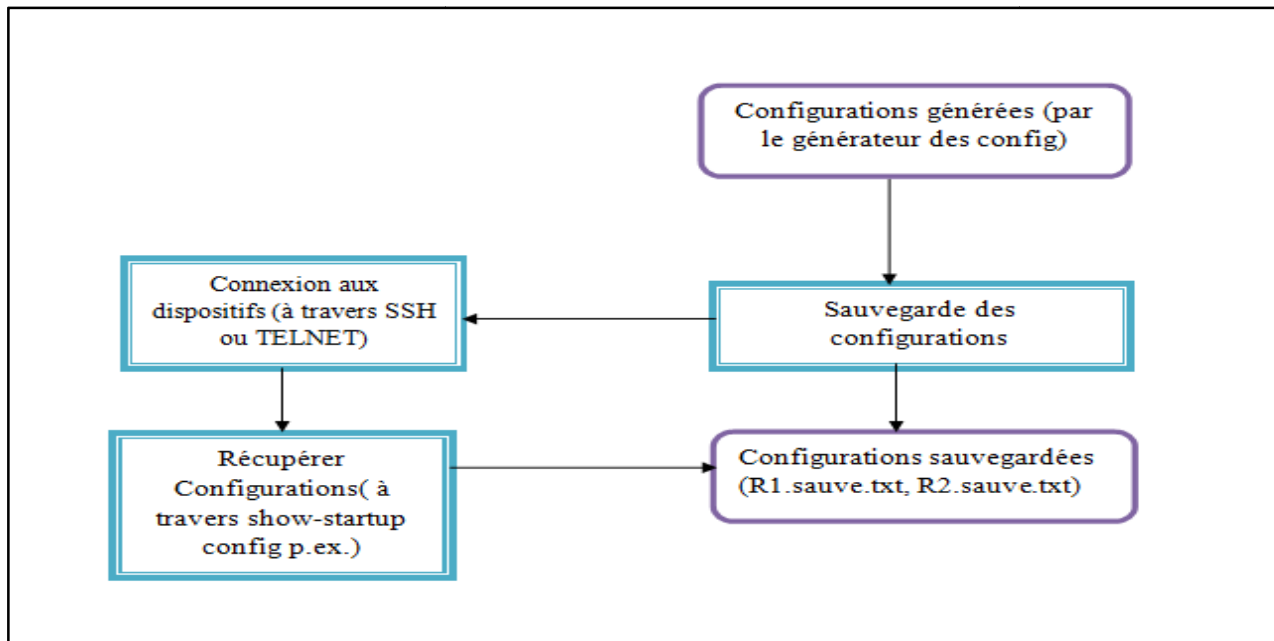


**Figure 7 : Module Générateur des configurations**

#### ➤ La sauvegarde des configurations

Ce processus a besoin d'une entrée : les configurations réseau. Il permet de les sauvegarder dans la base de données de l'application. Ces configurations sont obtenues de deux manières : soit par le processus de génération des configurations décrit plus haut, soit en passant par les modules de connexion aux dispositifs (à travers SSH ou TELNET) et de récupération des configurations (à travers l'exécution de certaines commandes, p.ex. show start-up config pour Cisco).

La figure suivante nous montre la structure du processus de sauvegarde des configurations :



**Figure 8 : Module sauvegarde des configurations**

### ➤ **Module sécurité des fichiers de configurations**

Il est important de sauvegarder les fichiers de configuration des différents équipements comme c'est démontré dans les sections plus haut. Mais il est encore plus important de sécuriser cette sauvegarde.

La sécurité des fichiers de configuration est un module de l'application GASC, qui permet de sécuriser ces fichiers de configurations. Ce module recouvre les deux aspects suivants :

- **la confidentialité** : il s'agit de s'assurer que les fichiers des configurations ne seront compréhensibles que par les administrateurs réseaux autorisés ;
- **l'intégrité** : il s'agit de s'assurer que les fichiers des configurations ne sont altérés que par les administrateurs réseaux autorisés ;
- **le contrôle d'accès** : il s'agit de garantir que seuls les utilisateurs autorisés auront accès aux fichiers des configurations.

- **La confidentialité**

La confidentialité est assurée par l'usage de l'algorithme de chiffrement symétrique AES [12].

GASC permet de chiffrer le contenu de la sauvegarde elle-même, afin de la rendre inaccessible à ceux qui n'y ont pas accès. Elle permet également de restaurer rapidement la sauvegarde grâce à l'opération de déchiffrement de cette dernière.

Pour assurer cette fonctionnalité GASC utilise le principe des algorithmes symétriques pour chiffrer et déchiffrer le contenu des fichiers de configurations.

Un algorithme symétrique est un algorithme qui permet de transformer un **texte en clair** en **texte chiffré** en utilisant une clé et de retransformer le **texte chiffré** en **texte en clair** en utilisant la même clé.

Le secret de la communication est uniquement assuré par la clé qui est utilisée lors de la phase de **chiffrement** et de **déchiffrement**. L'algorithme utilisé ne fait pas partie du secret.

On parle d'algorithmes symétriques car c'est la même **clé** qui sert à la fois au chiffrement et au déchiffrement du message.

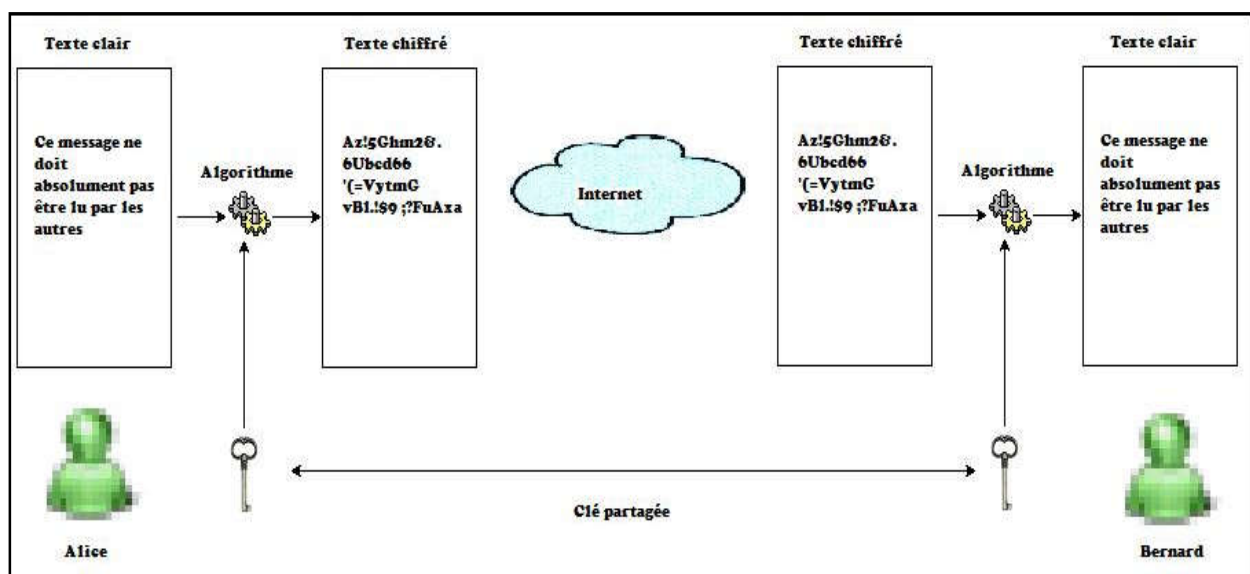


Figure 9 : Principe de l'algorithme symétrique [12]

GASC utilise plus particulièrement l'algorithme symétrique AES CBC avec une clé de 256 bits.

Le choix de l'algorithme AES dans GASC, se justifie par le fait que AES est un algorithme plus utilisé et le plus sûr de nos jours. De plus, son utilisation est très pratique car il consomme peu de mémoire, sa complexité est moindre et il est plus facile à mettre en œuvre.

- **L'intégrité**

La vérification de l'intégrité est assurée en utilisant un algorithme de hachage. Ici, GASC utilise l'algorithme SHA 256.

Avant de chiffrer les données des fichiers de configuration, on va calculer une signature, et on ajoutera ce hash aux données à chiffrer. Pour vérifier l'intégrité, il suffira alors, lors du

déchiffrement, d'extraire le hash des données, puis de recalculer le hash des données déchiffrées et de comparer les deux. S'ils sont égaux, alors les données sont correctes.

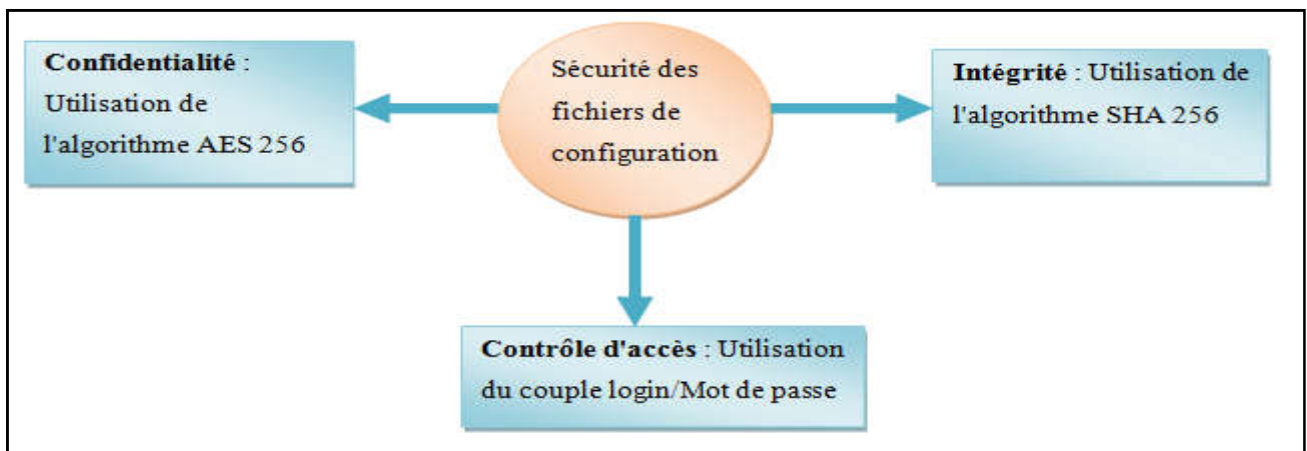
- **Le contrôle d'accès**

Le contrôle d'accès aux fichiers des configurations est assuré dans GASC par le couple login/mot de passe. Basée sur la gestion d'un secret, l'authentification par identifiant et mot de passe est un moyen simple et peu coûteux à déployer pour contrôler un accès. Toutefois, cette méthode d'authentification présente un niveau de sécurité faible.

Pour à la fois utiliser cette méthode et garantir un niveau de sécurité fiable, nous avons défini certaines exigences :

- longueur et complexité du mot de passe (huit caractères, utilisation des majuscules et caractères spéciaux)
- utiliser un algorithme pour chiffrer le mot de passe avant de stocker dans la base de données

La figure ci-après résume le module sécurité des fichiers des configurations :



**Figure 10 : Module sécurité des fichiers des configurations**

### 3.4.2 Conception détaillée

Dans cette partie, il s'agit de structurer les données, concevoir les IHM, détailler le fonctionnement des fonctionnalités.

#### 3.4.2.1 Structuration des données

Les données dans GASC, sont structurées de deux manières :

- les fichiers : contiennent les configurations des équipements réseaux ;
- une base de données : contient l'ensemble des données manipulées par l'application.



## Schéma final de la base de données

Le schéma de la base de données relationnelle est déduit à partir du diagramme des classes entités.

Le passage du diagramme des classes vers la base de données relationnelle se fait selon les règles de passage décrites ci-dessous :

- chaque classe entité est représentée par une table ;
- chaque attribut primitif donne une colonne dans la table ;
- chaque instance de la table est représentée par une ligne dans celle-ci ;
- la colonne de clé primaire est l'identificateur unique de l'instance ;
- chaque association « un à plusieurs » est représentée par une clé étrangère dans la table fille ;
- chaque association « plusieurs à plusieurs » entre deux classes est représentée par une nouvelle table qui prend comme clé primaire la concaténation des clés primaires des deux classes ;

chaque association « un à un » est représentée par l'intégration d'une clé étrangère dans la table la moins récente.

**Tableau 9: la table nœuds**

Nœuds				
Attributs	Type	Taille	Libellé	Contraintes
<b>Id</b>	Entier	10	Identifiant du nœud	Clé primaire
<b>Nom_routeur</b>	Texte	255	Nom du nœud	Obligatoire
<b>Constructeur</b>	Texte	255		
<b>Model</b>	Texte	10	Modèle du nœud	
<b>Version_os</b>	Texte	255	version de son système	
<b>Type</b>	Texte	50	Type du nœud : Routeur ou swtich	
<b>Username</b>	Texte	50	Nom d'utilisateur	
<b>Password</b>	Texte	255	Mot de passe utilisateur	
<b>enablePassword</b>	Texte	255	Mot de passe du mode enable	
<b>Cryptage</b>	Entier	2	Entier qui est à 1 si le	

Date_ajout	cryptage est activé 0 sinon	
	DATE	Date d'ajout du nœud

**Tableau 10 : table topologie**

topologie				
Attributs	Type	Taille	Libellé	Contraintes
Id	Entier	10	Identifiant de la topologie	Clé primaire
As_id	Entier	10	Numéro du domaine de la topologie	
Id_noeud	Texte	255	Identifiant du nœud	
Id_interface	Texte	255	Identifiant de l'interface du nœud	
Id_lien	Entier	10	Identifiant du lien	

**Tableau 11 : table utilisateurs**

utilisateurs				
Attributs	Type	Taille	Libellé	Contraintes
Id	Entier	10	Identifiant de l'utilisateur	Clé primaire
Nom	Texte	255	Nom de l'utilisateur	Obligatoire
Prenom	Texte	255	Prénom de l'utilisateur	
Mot_de_passe	Texte	255	Mot de passe de l'utilisateur	
Email	Texte	255	Adresse mail de l'utilisateur	
Statut	Texte	50	Statut de l'utilisateur (admis, etc.)	
Photo	Texte	255	Chemin qui mène vers la photo de l'utilisateur	
Timestamp	Entier	50	Temps de connexion de l'utilisateur	
Date_ajout	DATE		Date d'ajout de	

	l'utilisateur
--	---------------

**Tableau 12 : table configurations**

Configurations				
Attributs	Type	Taille	Libellé	Contraintes
<b>Id</b>	Entier	10	Identifiant de la config	Clé primaire
<b>Id_noeud</b>	Texte	255	Identifiant du nœud	Obligatoire
<b>Config_file</b>	Texte	255	Chemin complet vers le fichier de config	
<b>Config_file_name</b>	Texte	255	Nom du fichier de config	
<b>Hash_fichier</b>	Texte	255	Contient le hash du fichier de config	
<b>Date_ajout</b>	DATE		Date d'ajout de la config	

**Tableau 13 : table historiques**

Configurations				
Attributs	Type	Taille	Libellé	Contraintes
<b>Id</b>	Entier	10	Identifiant de l'action	Clé primaire
<b>Id_user</b>	Entier	10	Identifiant de l'utilisateur	Obligatoire
<b>Action</b>	Texte	255	L'action menée par l'utilisateur dans le système	
<b>Date_ajout</b>	DATE		Date d'ajout de la config	

**Tableau 14 : table taches**

taches				
Attributs	Type	Taille	Libellé	Contraintes
<b>Id</b>	Entier	10	Identifiant de la tâche	Clé primaire
<b>Nom</b>	Texte	255	Nom de la tâche	Obligatoire
<b>Description</b>	Texte	255		
<b>Crontime</b>	Texte	255	Durée de la cron (ex :	

	*/15)		
<b>Croncmd</b>	Texte	255	La commande à exécuter
<b>Date_ajout</b>	DATE		
<b>Ajouter_par</b>	Texte	255	Nom de l'administrateur
<b>Id_noeud</b>	Texte	255	Identifiants des nœuds concernés par cette tâche

**Tableau 15 : table liens**

liens				
Attributs	Type	Taille	Libellé	Contraintes
<b>Id</b>	Entier	10	Identifiant du lien	Clé primaire
<b>Id_noeud1</b>	Texte	255	Identifiant du premier nœud	Obligatoire
<b>Id_interface1</b>	Texte	255	Identifiant de l'interface du nœud 1	
<b>Id_noeud2</b>	Texte	255		
<b>Id_interface2</b>	Texte	255		
<b>Mtu</b>	Entier	10	Maximum transmit unit	
<b>Speed</b>	Entier	10	Bande passante	

**Tableau 16 : table interfaces**

interfaces				
Attributs	Type	Taille	Libellé	Contraintes
<b>Id</b>	Entier	10	Identifiant de l'interface	Clé primaire
<b>Id_interface</b>	Texte	255	Nom de l'interface	Obligatoire
<b>Id_noeud1</b>	Texte	255	Nom du nœud	Obligatoire
<b>Ip</b>	Texte	255	Adresse IP de l'interface	
<b>Mask</b>	Entier	10	Masque de l'IP	
<b>Mtu</b>	Entier	10	Maximum transmit unit	

**Tableau 17 : table domaine**

domaine				
Attributs	Type	Taille	Libellé	Contraintes
Id	Entier	10	Identifiant du domaine réseau	Clé primaire
As_id	Entier	10	Numéro de l'autonomous System	Obligatoire
Nom	Texte	255	Nom du domaine	Obligatoire

**Tableau 18 : table routes\_statiques**

routes_statiques				
Attributs	Type	Taille	Libellé	Contraintes
Id	Entier	10	Identifiant de la route	Clé primaire
Id_routeur	Entier	10	Identifiant du routeur	Obligatoire
Ip_destination	Texte	255	Adresse IP de la destination	Obligatoire
Mask	Entier	10	Masque de l'IP	
Ip_next_hop	Texte	255	Adresse IP du prochain routeur	

### 3.4.2.2 Conception des IHM

Dans cette section, il est question de donner la charte graphique que devra respecter toutes les différentes interfaces de l'application. Cette charte graphique contient trois grandes parties : **Le menu, l'entête et la partie Traitement**. Elle est donnée par la figure suivante :

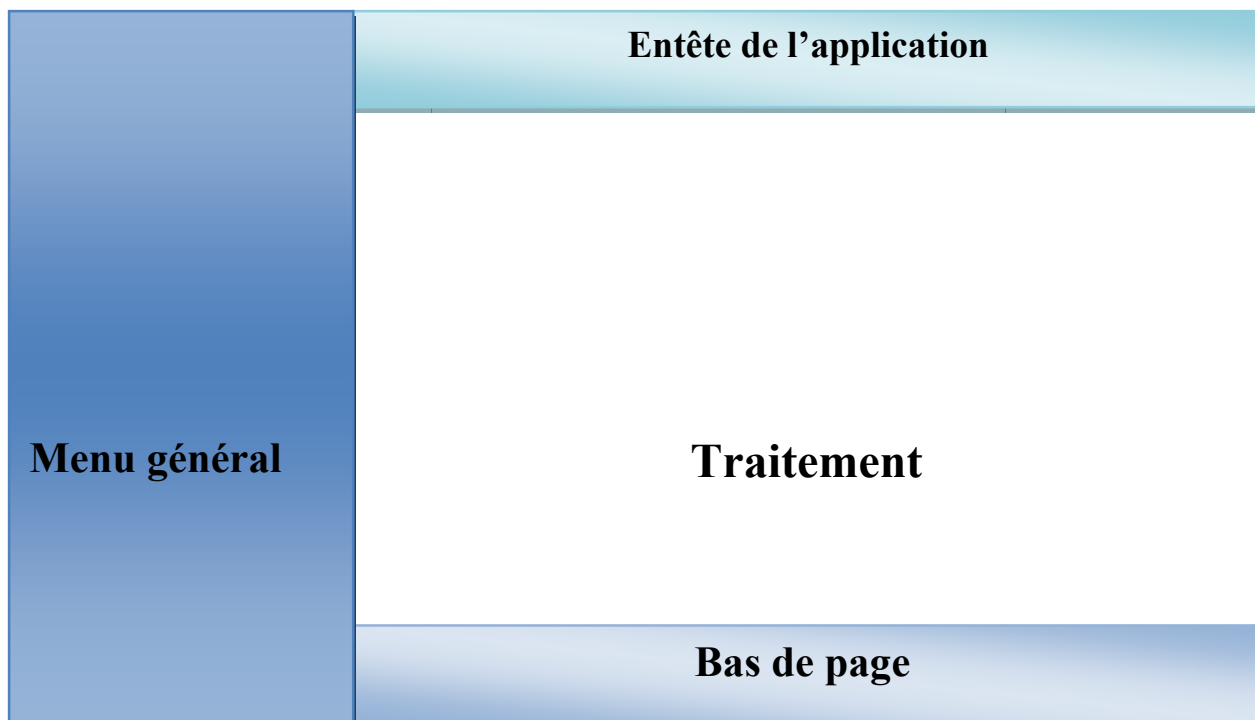


Figure 11 : Charte graphique de GASC

### 3.4.2.3 Description détaillée des fonctionnalités

#### a. Diagramme de séquence « Authentification »

Dans cette section, nous donnerons une description détaillée des Diagrammes de séquences pour quelques cas d'utilisation les plus importants.

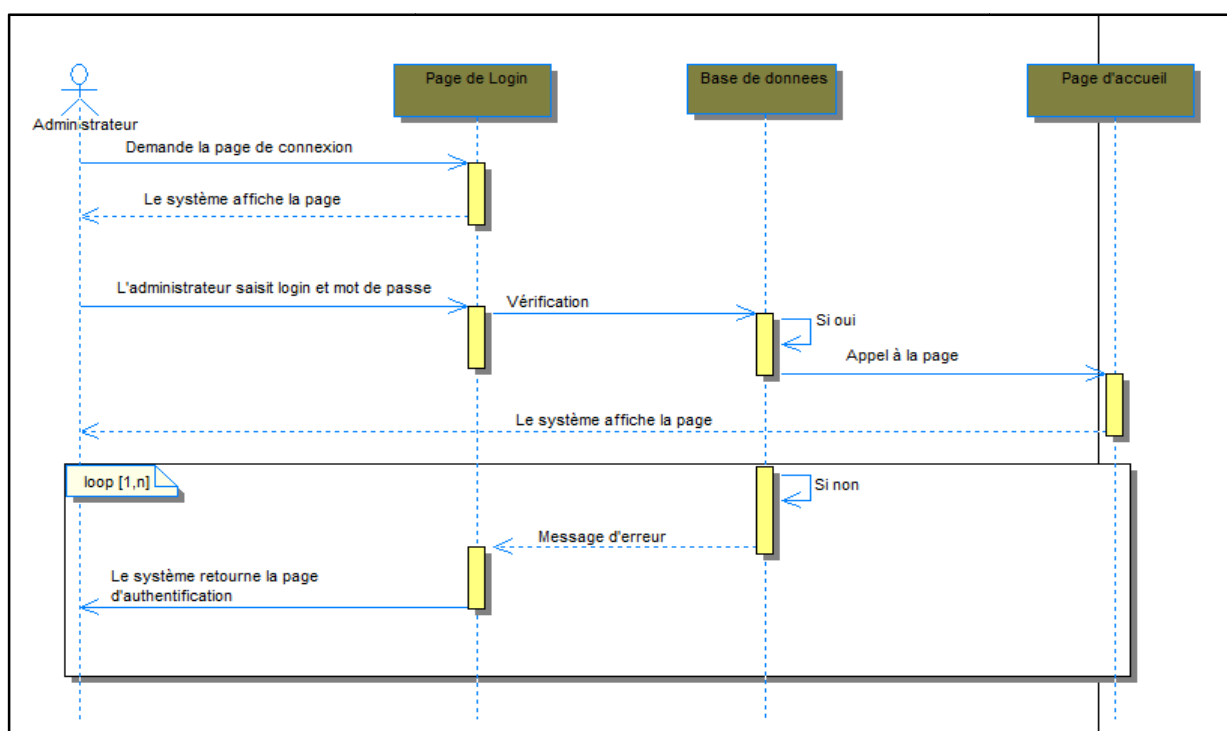
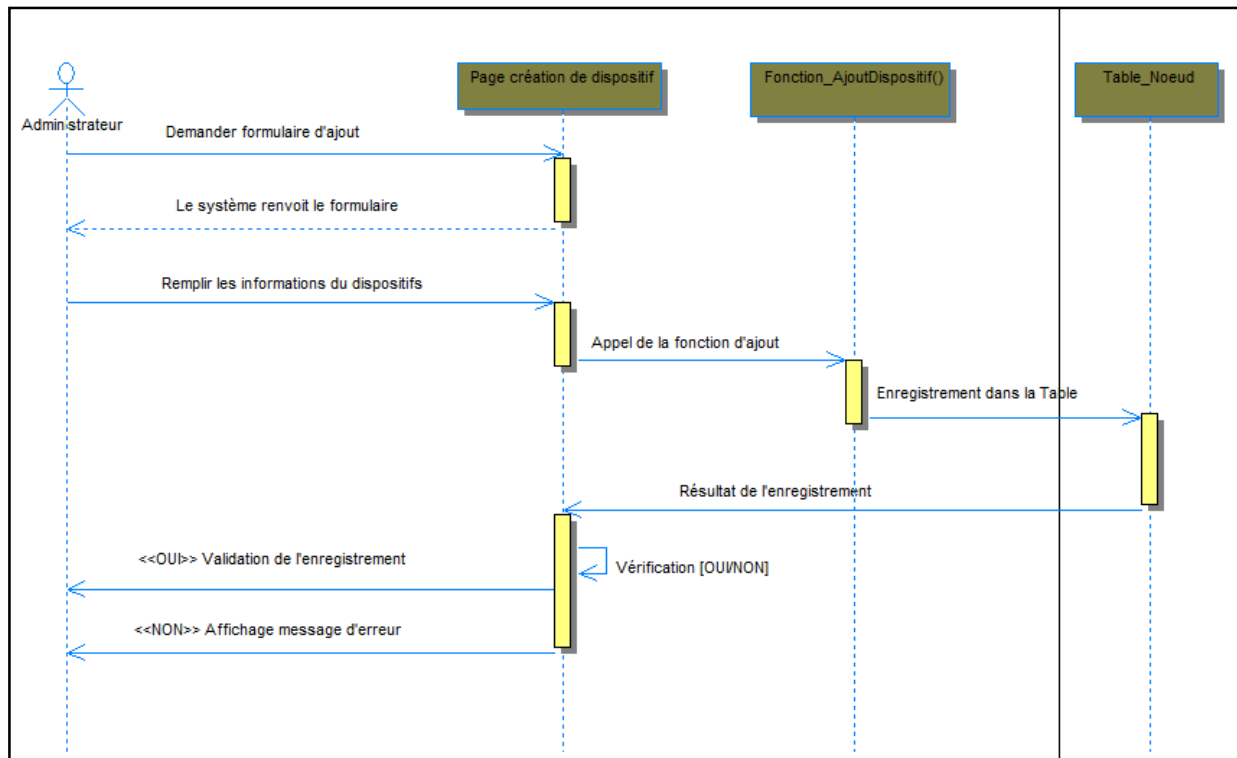


Figure 12 : Diagramme de séquence d'authentification

Le diagramme de séquence « **Authentification** » présente le séquencement des interactions entre **l'utilisateur, l'interface d'authentification, l'entité base de données et la page de l'utilisateur**. Dans ce diagramme **loup (1, n)** indique qu'il y aura une répétition d'affichage de l'interface d'authentification jusqu'à la validation du nom et le mot de passe de l'utilisateur.

#### b. Diagramme de séquence « Ajouter un dispositif »



**Figure 13 : Diagramme de séquence Ajouter un dispositif**

Le diagramme de séquence d'ajout d'un nouveau dispositif présente le séquencement des interactions entre l'administrateur, la page de création d'un dispositif et la table **Nœuds**.

#### c. Diagramme de séquence «Créer la topologie »

Le diagramme de séquence de création de la topologie présente le séquencement des interactions entre l'administrateur, la page de création de la topologie et la table **topologie**.

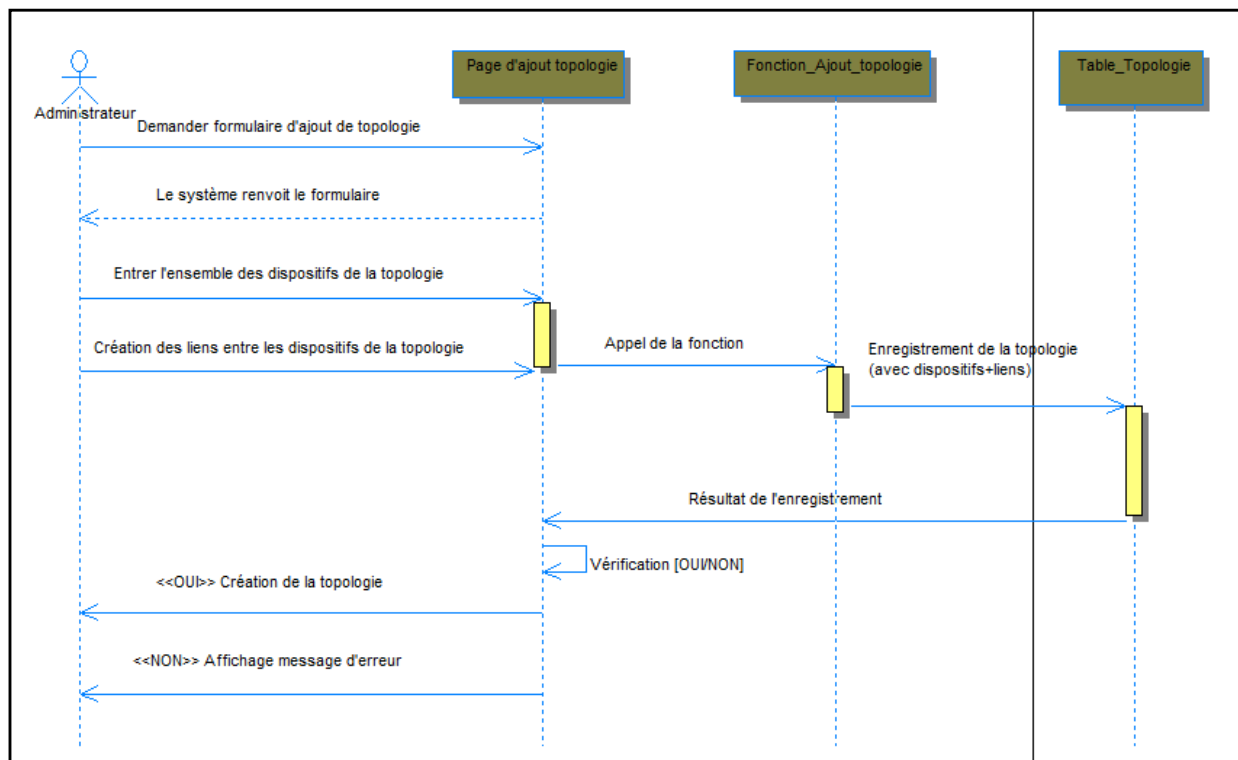


Figure 14 : Diagramme de séquence Créer une topologie réseau

#### d. Diagramme de séquence «Comparer deux archives »

Le diagramme de séquence de Comparaison des archives présente le séquençement des interactions entre l'administrateur, la page de comparaison (un formulaire) et la fonction **CompareConfiguration**.

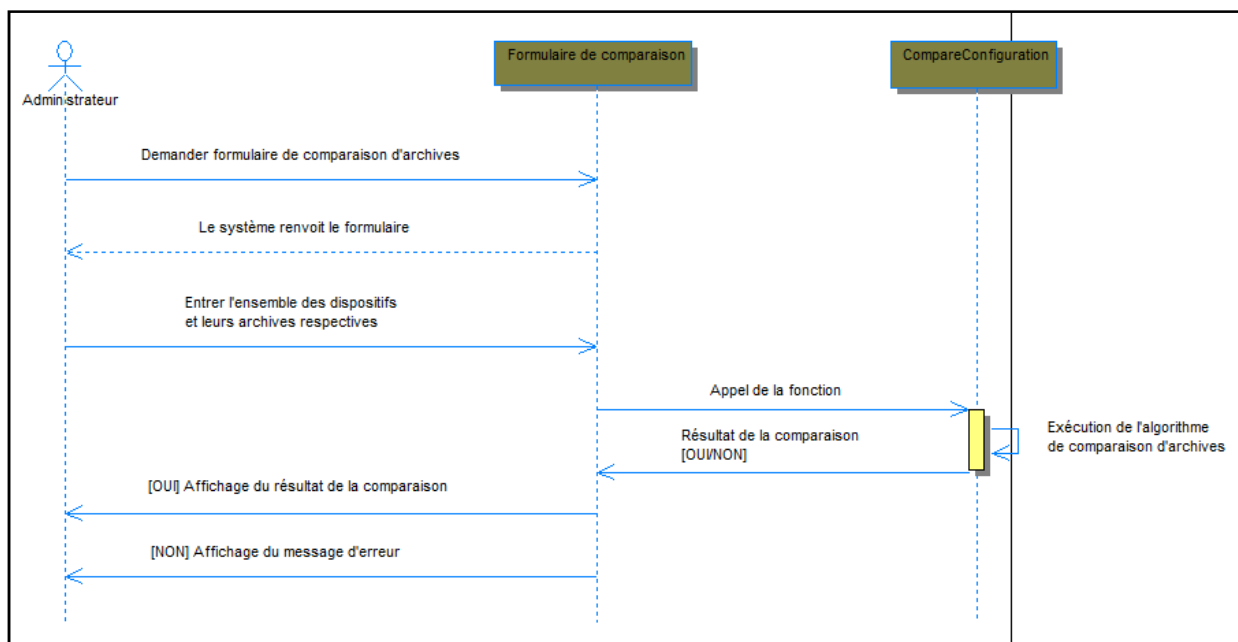


Figure 15 : Diagramme de séquence «Comparer deux archives »



### 3.5 Conception des algorithmes

Dans cette partie, nous allons décrire les différents algorithmes utilisés par l'application GASC.

#### 3.5.1 Algorithme de génération des configurations

**Input :** Topologie valide = <Adresses IP ; Dispositifs ; Liens>

**Input :** format des configurations = <cisco.conf ; juniper.conf>

**Output :** Configurations générées en fichier .Txt = <router.txt>

1. **Variable** noeuds\_topologie  $\leftarrow$  L'ensemble des nœuds de la topologie ;
2. **Variable** i  $\leftarrow$  1;
3. **Tant que** (i < noeuds\_topologie ) **Faire**
4.     charger\_format\_des\_configurations (i) ;
5.     Afficher\_configurations (i) ;
6. **Fin Tant que**

Figure 16 : Algorithme de génération des configurations

#### 3.5.2 Algorithme de sauvegarde des configurations

**Input :** Adresse IP = <192.168.2.7>

**Input :** méthode de connexion = <SSH ; TELNET>

**Input :** Port de connexion = <21, 22 , 23>

**Input :** Username, mot de passe

**output :** Configurations sauvegardées en fichier .Txt = <router.txt>

```

1. Variable commande_a_exécuter ← " p.ex. show start-up config " ;
2. Variable NombreNoeud , i ;
3. Tant que (i < NombreNoeud ) Faire
4.     connexion_au_noeud (i) ;
5.     executer_commande (commande_a_exécuter ) ;
6.     Recuperer_donnees_config (i);
7.     if config_exist (i) Then
8.         H1 = Calculer_hash_nouveau_fichier ();
9.         H2 = Calculer_hash_ancien_fichier ();
10.        if (H1 = H2) Then
11.            Ne rien Faire.
12.        else
13.            Stocker_fichier_dans_la_base_de_donnees ();
14.    end
15. Fin Tant que

```

**Figure 17 : Algorithme sauvegarde des configurations**

### 3.5.3 Synoptique de l'algorithme de sauvegarde des configurations

La figure suivante montre le synoptique de l'algorithme de sauvegarde des configurations :

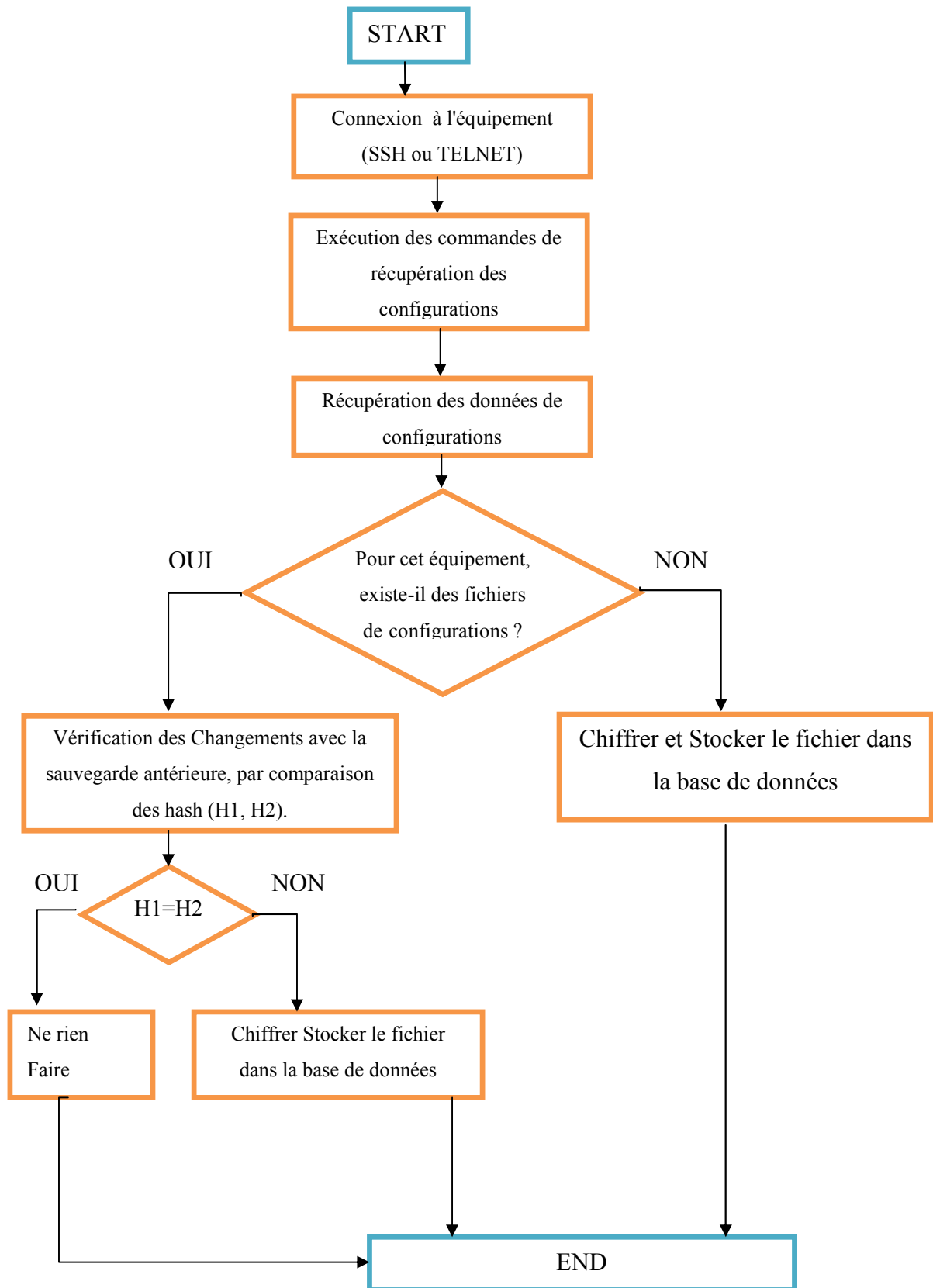


Figure 18 : Synoptique de l'algorithme de sauvegarde des configurations

Cette partie a été consacrée à la modélisation de l'aspect statique du système, en se basant sur les spécifications détaillées du système. Dans le prochain chapitre on aborde la partie mise en œuvre ainsi que la description de quelques choix techniques effectués pendant le développement.

## CHAPITRE 3 : MISE EN ŒUVRE ET RESULTATS

Ce chapitre sera consacré au choix des outils utilisés pour la réalisation de l' application, à la réalisation des fonctionnalités ainsi qu'aux principales interfaces de l'application.

### 3.1 Choix des outils

#### 3.1.1 Plateforme de développement : WampServer

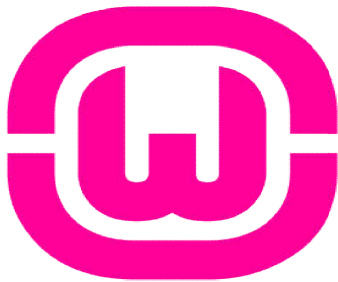


Figure 19 : Wamp server [13]

Wamp server est une plateforme de développement des applications web dynamiques. Ce logiciel est très intéressant puisqu'il englobe tous les outils nécessaires pour le fonctionnement d'une application web notamment un serveur de base de données MySQL, un serveur web apache et une interface de gestion des bases de données facile à utiliser **PHPMysqladmin**. [13]

#### 3.1.2 Editeur de code source : Sublime Text 3



Figure 20 : Sublime Text 3 [14]

Sublime Text est un éditeur de code développé en C++ et Python. Il supporte de nombreux langages de programmation et ses fonctionnalités peuvent être étendues grâce à ses plugins. Multiplateformes, il est disponible sur **Linux, Mac et Windows**. [14]

Tableau 19: Langages de programmation utilisés

LANGAGE	DESCRIPTION
HTML 5	Le langage HTML est la base de nombreuses technologies Web. Il est composé de balises qui permettent de décrire l'affichage d'une page. <b>Il est utilisé dans le premier tiers de l'architecture de la solution.</b>
	<b>CSS</b> est un langage dont l'objectif est de paramétrer finement la mise en page et la mise

CSS 3	en forme des éléments d'un document HTML. C'est un langage interprété par le navigateur web. <b>Il est utilisé dans le premier tiers de l'architecture de la solution.</b>
BOOTSTRAP 3.0	<b>Bootstrap</b> est un Framework qui peut rendre la tâche des développeurs plus facile pour créer l'architecture d'une page web. <b>Il est utilisé dans le premier tiers de l'architecture de la solution.</b>
JavaScript/AJAX	Le principe de cette technologie est d'ajouter au code HTML, un certain nombre de fonctionnalités qui pourront être interprétées par le navigateur en réponse à certains événements (clic de l'utilisateur, survol d'une zone etc.). <b>Il est utilisé dans le premier tiers de l'architecture de la solution.</b>
PHP 5	En complément à HTML, l'utilisation du langage PHP permet à un serveur de générer du code HTML dynamiquement en réponse à une requête d'un client. <b>Il est utilisé dans le second tiers de l'architecture de la solution.</b>

### 3.1.3 Système de gestion de base des données : MySQL

Notre choix s'est porté sur **MySQL**. C'est un système de gestion de base de données (SGBD) qui fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels.

## 3.2 Mise en place de l'environnement de développement

Pour mettre en place l'environnement de développement, nous avons utilisé un PC qui a les caractéristiques suivantes :

- PC portable TOSHIBA ;
- Processeur Pentium (R) Dual-core 2.00 GHz \* 2;

- Mémoire installée : 6 GB ;
- Version PHP : 5.6.35 ;
- Version apache : 2.4.17 ;
- Version MySQL : 5.7.21 ;
- Système d'exploitation : Windows 8.1 Professionnel.

Sur cette machine, nous avons d'abord installé le logiciel Wamp Server décrit plus haut. Une description de l'installation est donnée dans l'annexe B.

### 3.3 Réalisation des fonctionnalités

Dans cette section, il s'agit d'expliquer comment certaines fonctionnalités de l'application les plus complexes ont été réalisées. Pour cela, nous allons décrire les fonctionnalités suivantes :

- Générateur des configurations ;
- Sauvegarde des configurations ;
- Sécurité des fichiers des configurations ;
- Comparateur des fichiers de configurations.

#### 3.3.1 Générateur des configurations

Pour réaliser cette fonctionnalité, nous avons d'abord commencé par écrire des requêtes SQL, qui permettent de récupérer l'ensemble des nœuds de la topologie avec leur différents paramètres :

- **nom du constructeur** : sert à retrouver le Template des configurations adéquat ;
- **numéro de la version de l'IOS** : certaines versions des équipements ne supportent pas des commandes récentes, ce paramètre permet à l'application d'adapter les commandes ;
- les interfaces avec leurs adresses IP ;
- toutes les liaisons entre les nœuds.

Par la suite nous avons effectué une boucle PHP (**foreach**) qui permet pour chaque nœud récupéré précédemment, d'aller chercher le Template des configurations adéquat, de produire les configurations dans une zone d'affichage (**textarea en HTML5**), puis de passer ces configurations au module de sauvegarde à fin de les chiffrer et les stocker dans la base de données.

### 3.3.2 Sauvegarde des configurations

Les configurations pour la sauvegarde sont récupérées de deux manières :

- **A partir du générateur des configurations décrits plus haut** : dans ce cas le module sauvegarde des configurations effectuent certaines opérations telles que la création du fichier de configuration, la récupération de la date de génération de la configuration, le chiffrement du fichier et le stockage dans la base de données.
- **A partir d'une connexion établie par SSH ou TELNET sur le nœud** : dans ce cas le module sauvegarde des configurations récupère les paramètres de connexion du nœud (nom d'utilisateur, mot de passe et adresse IP), effectue la connexion puis exécute sur le nœud les commandes nécessaires pour le rapatriement des configurations. Une fois que le nœud répond à ces commandes par un flux de données des configurations, ce module reprend exactement les opérations effectuées pour le premier cas ci-dessus.

### 3.3.3 Comparateur des fichiers de configurations

Parfois, les modifications de configuration ne sont apportées qu'à la configuration en mémoire. Toutefois, si l'appareil tombe en panne, ces modifications sont perdues. C'est pourquoi il est important de sauvegarder également les modifications de configuration permanentes dans la configuration de démarrage.

Le module Comparateur des fichiers est utilisé pour comparer la configuration en cours avec la configuration de démarrage, de cette façon, nous pouvons facilement via une interface graphique identifier les modifications qui n'ont pas été enregistrées.

Nous avons réalisé ce module en deux étapes :

- **première étape** : ici, le module récupère les contenus des deux fichiers (grâce à la fonction *file\_get\_contents()* de PHP) et découpe ces contenus en segments (lignes) grâce à la fonction *explode* de PHP.
- **seconde étape** : dans cette étape, le module compare les lignes des deux fichiers trouvées dans la première étape et affiche la différence dans une zone d'affichage suivant un code de couleur :
  - Rouge : Couleur pour ligne "supprimée" ;
  - Vert clair : Couleur pour ligne "ajoutée" ;
  - Blanc : Couleur pour ligne "normale".



## 3.4 Déploiement

### 3.4.1 Diagramme de déploiement

Dans cette partie nous allons décrire l'implantation physique de l'application GASC grâce à un diagramme proposé par UML : le diagramme de déploiement.

Le diagramme de déploiement permet de représenter l'architecture physique supportant l'exploitation du système. Cette architecture comprend des nœuds correspondant aux supports physiques (serveurs, routeurs...) ainsi que la répartition des artefacts logiciels (bibliothèques, exécutables...) sur ces nœuds. C'est un véritable réseau constitué de nœuds et de connexions entre ces nœuds qui modélise cette architecture.

- Un **nœud** correspond à une ressource matérielle de traitement sur laquelle des artefacts seront mis en œuvre pour l'exploitation du système. Les nœuds peuvent être interconnectés pour former un réseau d'éléments physiques.
- Un **artefact** est la spécification d'un élément physique qui est utilisé ou produit par le processus de développement du logiciel ou par le déploiement du système. C'est donc un élément concret comme par exemple : un fichier, un exécutable ou une table d'une base de données.

#### Formalisme et exemple

Un nœud ou une instance de nœud se représente par un cube ou parallélépipède (fig. suivante) :

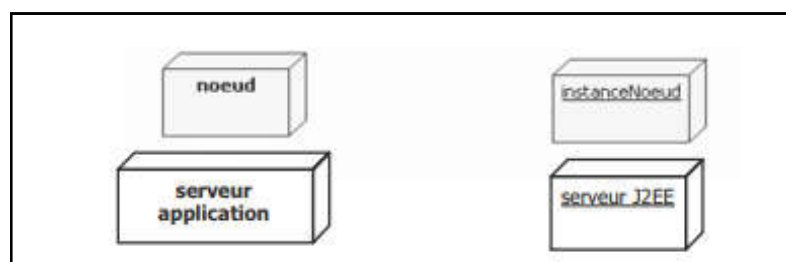
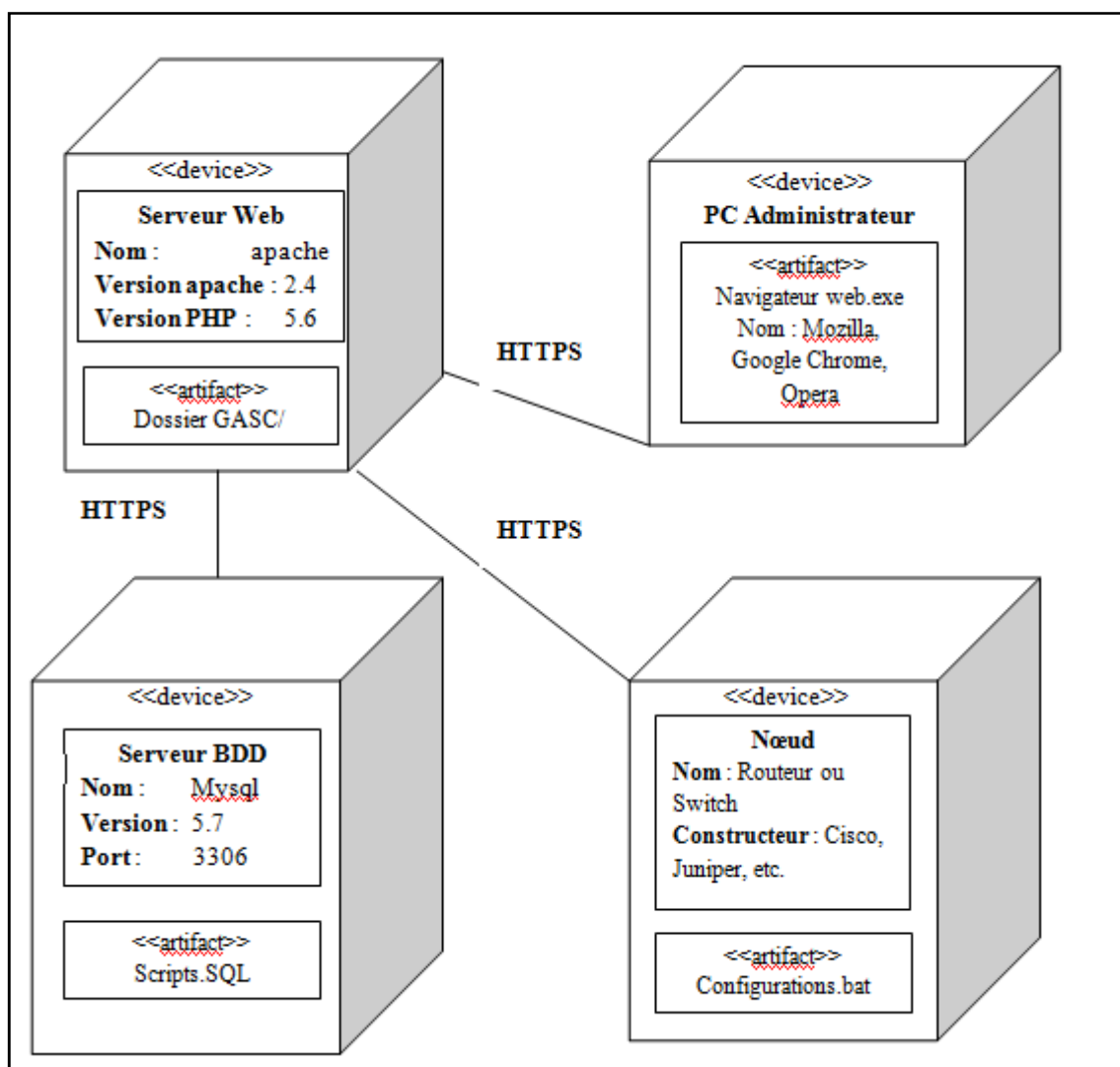


Figure 21 : Représentation de nœuds

La figure ci-après montre le diagramme de déploiement de la solution :



**Figure 22 : Diagramme de déploiement**

Dans la figure ci-dessus plusieurs composants sont déployés :

- Un serveur web où se trouve le code source PHP et les autres éléments de l'application dans un dossier nommé GASC : images, feuilles de style, pages html ;
- Un serveur BDD (base de données) sur lequel sont stockées des procédures stockées SQL : « scripts.sql » ;
- Un nœud qui peut être un routeur ou un Switch qui contient toutes les configurations réseaux : « configurations.bat » ;
- Un PC de l'administrateur qui permet d'accéder aux différentes ressources de l'application grâce au : « navigateur.exe ».

### 3.5 Résultats : quelques scénarii d'utilisation du système

Cette partie recense la présentation du Scénario applicatif de l'application, nous présentons, dans ce qui suit, les espaces d'utilisation. Pour accéder à l'application, l'utilisateur doit tout d'abord s'authentifier, pour cela il introduit son login et son mot de passe comme l'indique la figure suivante. Si les informations fournies sont valides, ce dernier est redirigé vers son propre espace.

Les pages qui suivent l'authentification contiennent tous des éléments communs. D'une part l'élément **header** (entête de l'application) qui contient le nom de l'acteur sous forme d'une liste, contenant deux boutons un pour se déconnecter et fermer la session et l'autre pour pouvoir modifier des données qui lui sont personnel(nom, email, adresse, etc. ). D'autre part un élément **sidebar**, c'est-à-dire un menu latéral, qui va permettre de naviguer dans les différentes pages de l'application.

#### 3.5.1 Processus d'authentification

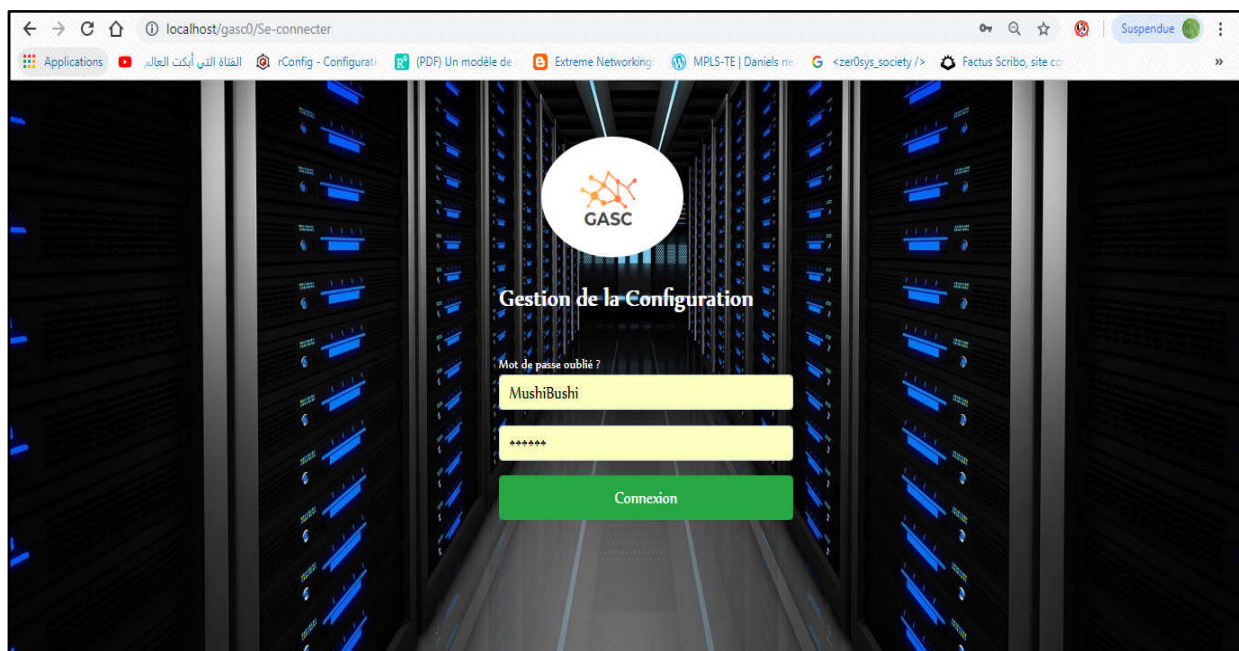


Figure 23 : Processus d'authentification

Une fois que l'utilisateur saisit son **nom et son mot de passe**, l'application procède à la vérification de ces informations. Si les informations sont validées par l'application, alors la page d'accueil s'ouvrira, dans le cas contraire un message d'erreur est affiché.

### 3.5.2 L'accueil de l'application

La page d'accueil se présente comme l'indique la figure ci-dessous, à travers laquelle l'utilisateur peut choisir une des options figurant dans le **menu à gauche**.

Dans la partie **1**, nous avons trois boutons, un 1<sup>er</sup> qui mène vers le paramétrage de l'application, un 2<sup>ème</sup> qui mène vers le profil de l'utilisateur et un dernier qui permet à ce dernier de se déconnecter de l'application. La partie **2** est réservée aux alertes émises par l'application pour indiquer par exemple qu'un nœud ne possède pas d'adresse IP. La partie **3** affiche les 5 derniers nœuds ajoutés dans l'application.

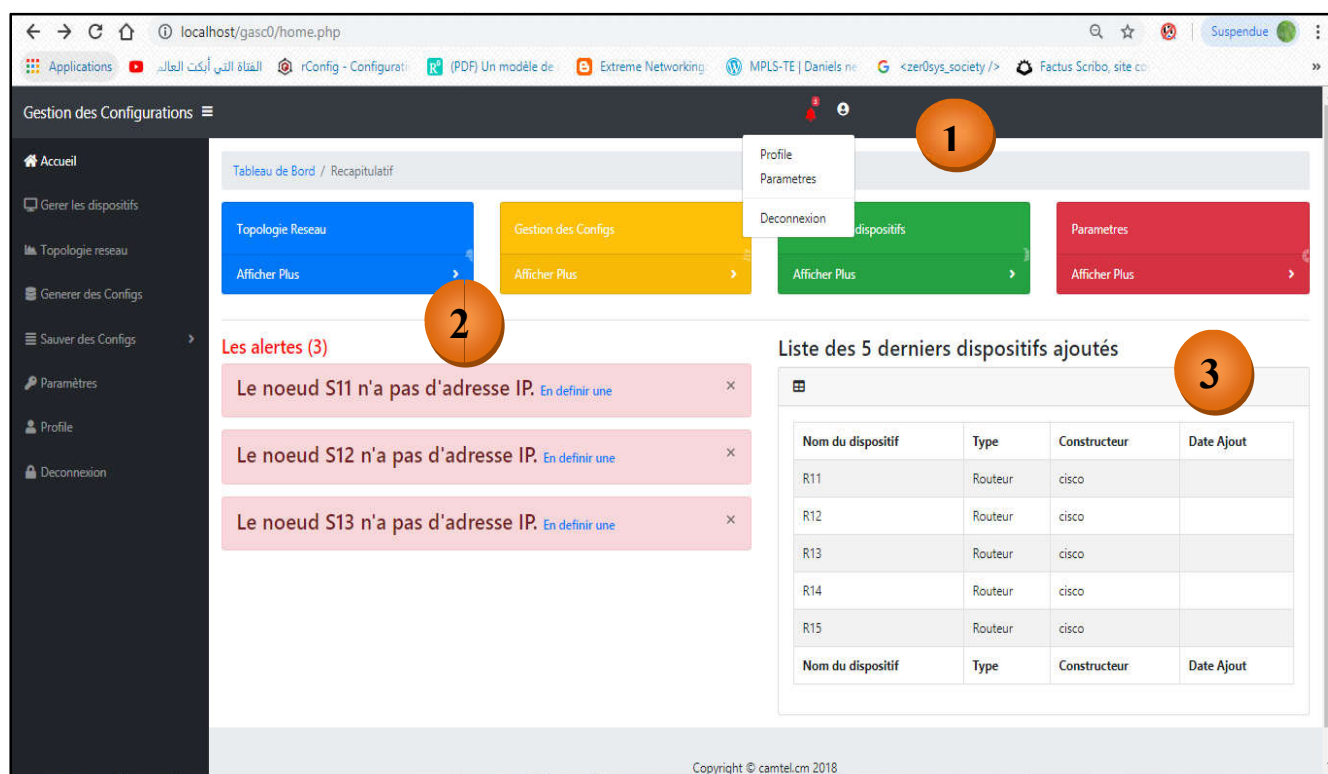


Figure 24 : L'accueil de l'application

### 3.5.3 Processus d'édition d'un nœud

Pour éditer un nœud, l'administrateur clique sur le lien "Gestion des dispositifs", un tableau s'affiche contenant la liste des nœuds. Il clique ensuite sur le bouton Editer et La page ci-dessous s'ouvre. Elle permet d'éditer les informations d'un nœud sélectionné. La partie **1** contient les informations générales sur le nœud (adresse IP, statut, constructeur, etc.). La partie **2** contient un bouton qui permet de lancer une page d'ajout d'une nouvelle interface. La partie **3** contient un tableau affichant les informations sur les interfaces du nœud (nom, adresse IP, masque, MTU, etc.). La partie **4** est une zone contenant la configuration courante du nœud.

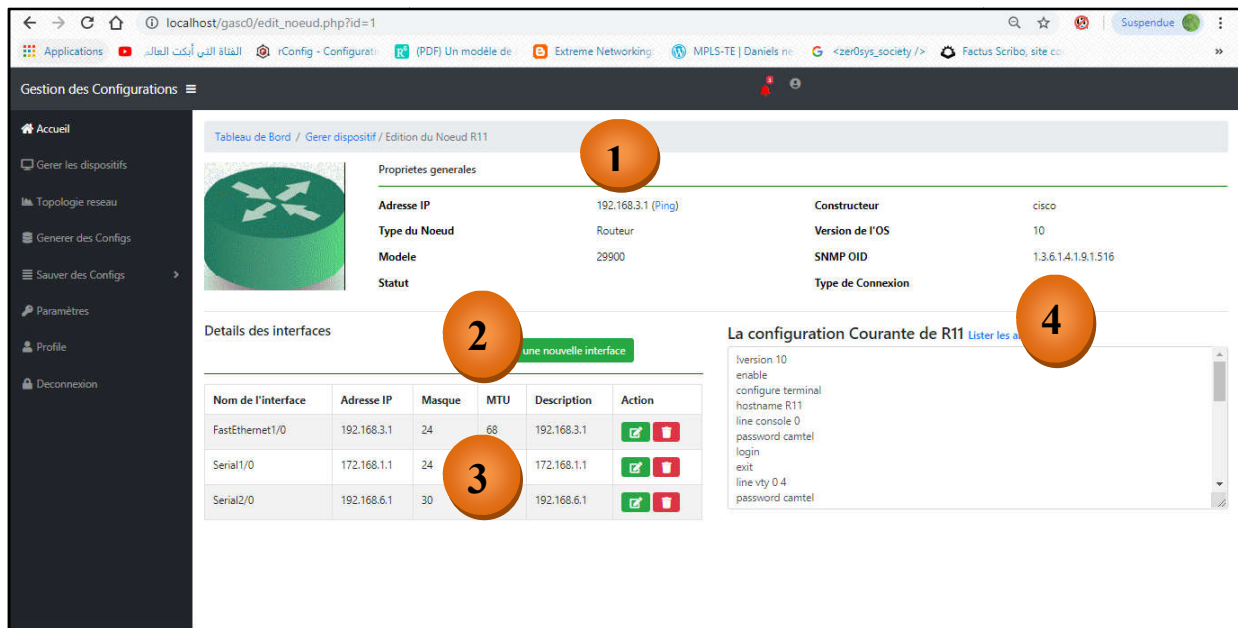


Figure 25 : Processus d'édition d'un nœud

### 3.5.4 Processus d'ajout d'une nouvelle interface

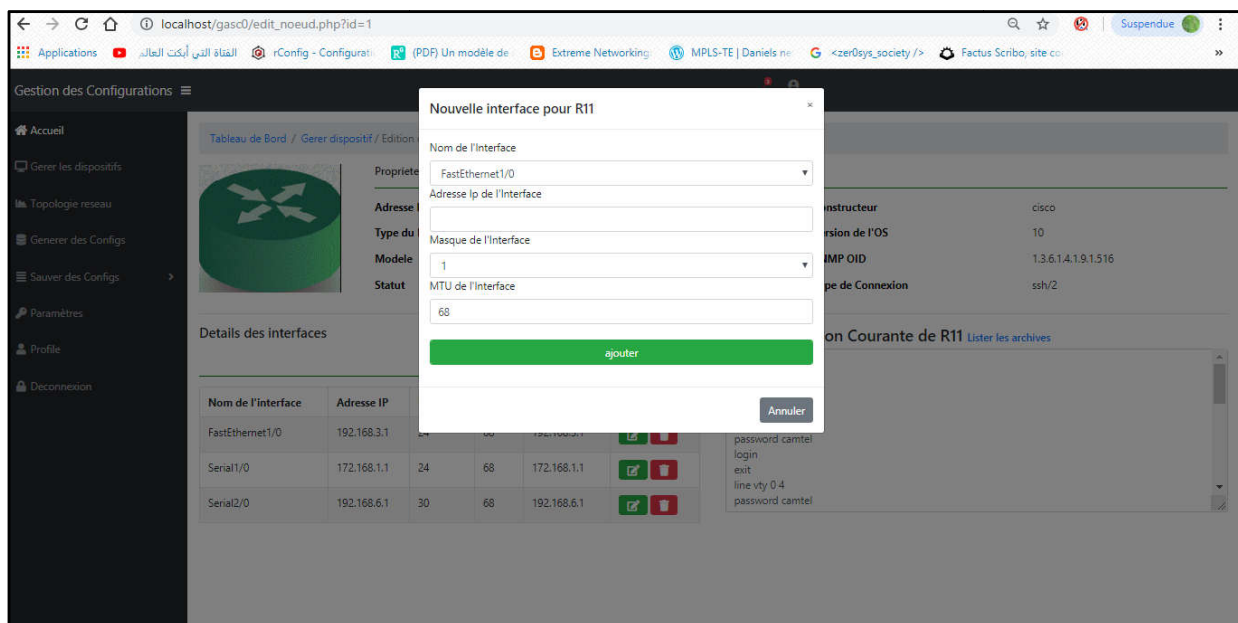


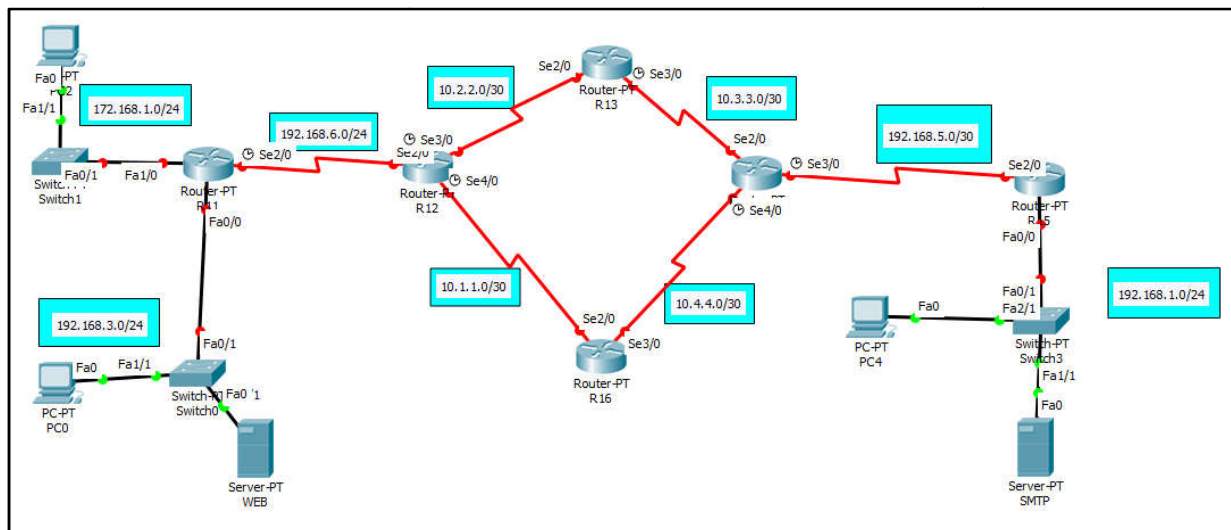
Figure 26 : Processus d'ajout d'une nouvelle interface

Pour configurer une interface d'un nœud, l'administrateur clique sur le bouton Editier du nœud pour le quel il souhaite configurer une interface qui apparaît sur le tableau des nœuds, la page du nœud en question s'ouvre, il clique ensuite sur le bouton "Ajouter une interface" et la figure ci-dessus apparaît qui contient un formulaire, il remplit les paramètres et clique sur "Ajouter".

### 3.6 Validation de la solution : Etude de cas réelle

Dans cette partie, nous présenterons une étude de cas basée sur une topologie réseau composée de six routeurs et trois Switchs que nous avons réalisé dans le logiciel de simulation **PACKET TRACER**. L'étude présentera de façon synthétique un exemple d'utilisation de notre application, qui permettra de générer les configurations d'interconnexion des différents équipements (Switchs et routeurs).

### 3.6.1 Topologie réseau



### Figure 27 : Topologie réseau

La façon dont la topologie est ajoutée dans GASC est donnée dans la (**figure 28 : gestion de la topologie**).

La figure **28** permet à l'administrateur de gérer la topologie réseau. La partie **1** permet d'ajouter des nœuds à la topologie. Une fois ces nœuds ajoutés, l'administrateur créera les liens entre tous les nœuds de la topologie. La partie **2** contient un bouton qui permet de renvoyer vers la page de génération de configuration.

### 3.6.2 Processus d'ajout de topologie

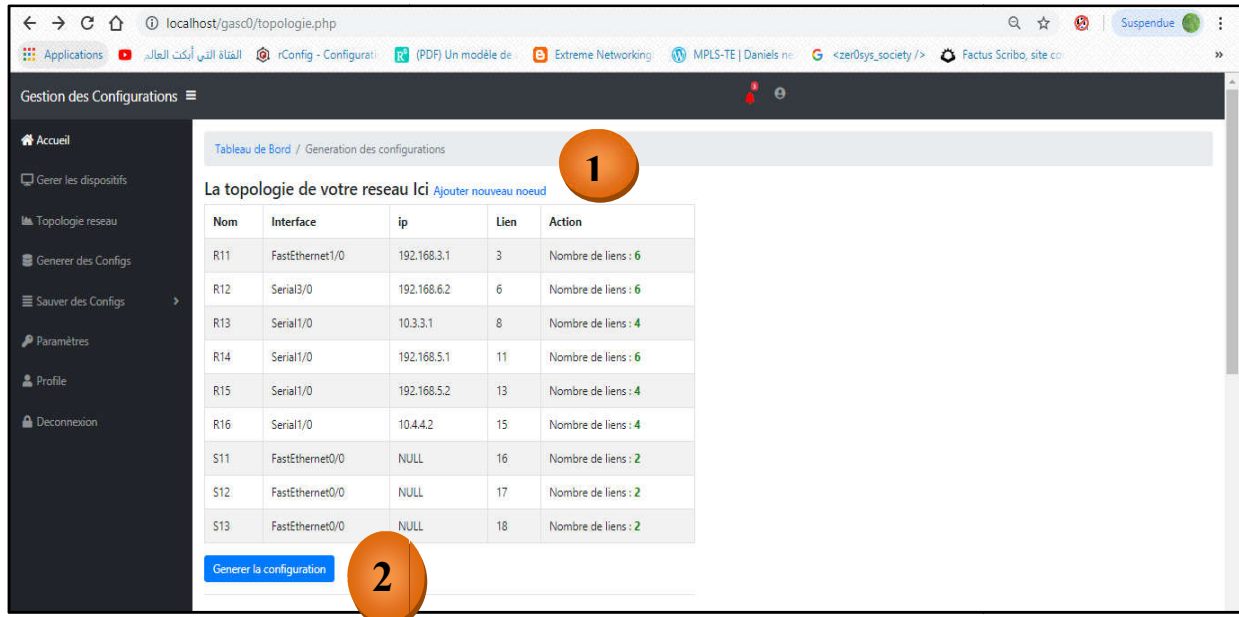


Figure 28 : Processus d'ajout de topologie

### 3.6.3 Création des liens

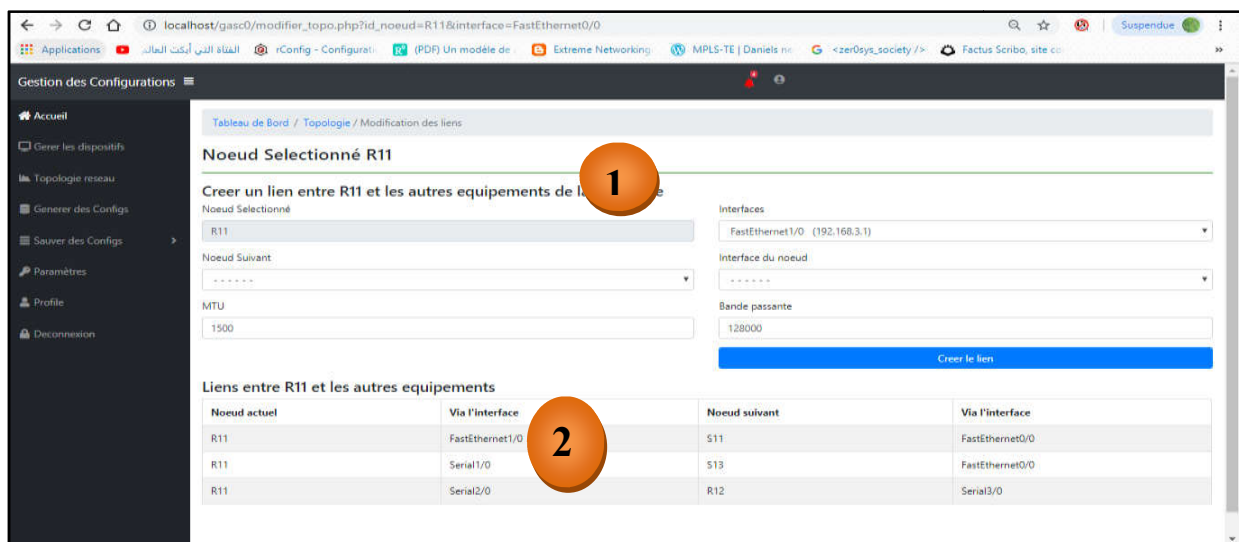


Figure 29 : Création des liens dans GASC

La figure 29 montre comment les liens sont créés dans l'application GASC. La parti 1 contient le formulaire de création des liens. La partie 2 affiche tous les liens du nœud sélectionné.



### 3.6.4 Génération des configurations

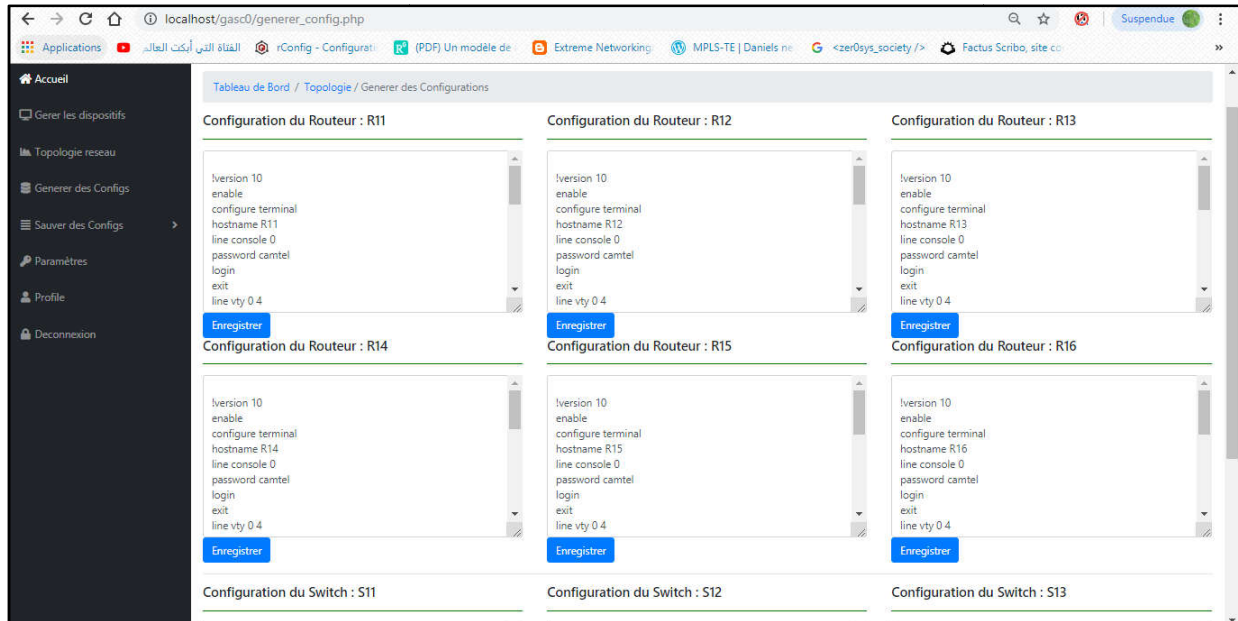


Figure 30 : Génération des configurations

La figure 30 montre les différentes configurations générées pour les équipements de la topologie présentée dans l'étude de cas plus haut. Par une simple opération de copier coller, l'administrateur pourra configurer tous les équipements. En cliquant sur le bouton Enregistrer, les configurations sont sauvegardées dans la base de données.

Ci-après nous avons capturé les résultats de la commande "ping" avant (ECHEC de la commande) et après (REUSSITE) la génération des configurations par l'application GASC :

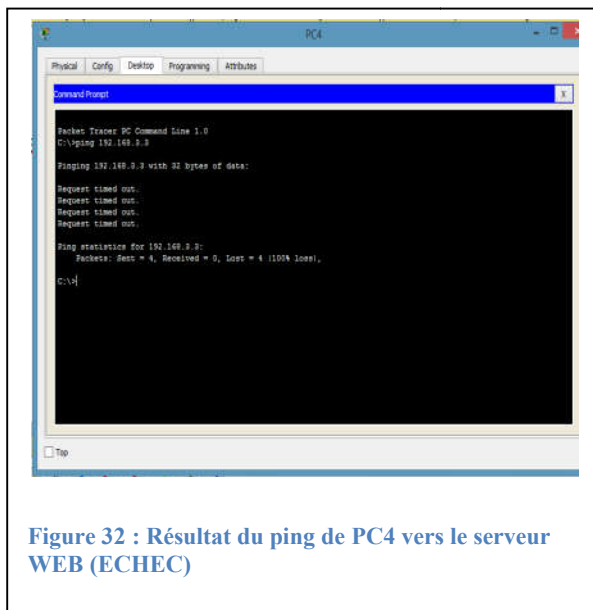


Figure 32 : Résultat du ping de PC4 vers le serveur WEB (ECHEC)

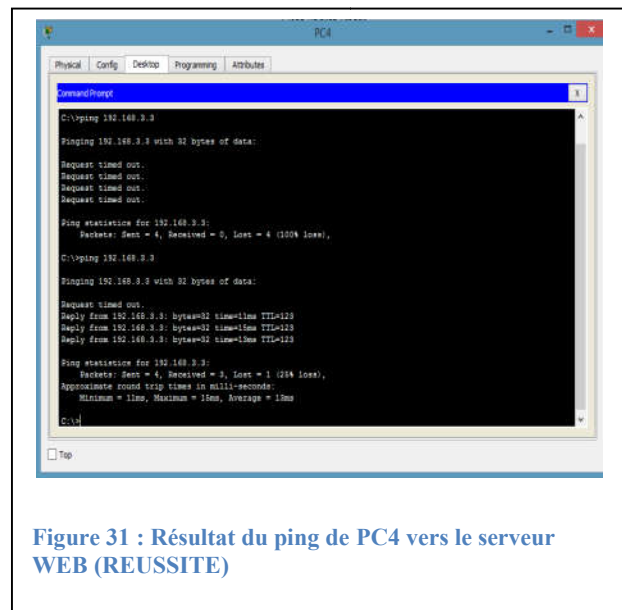


Figure 31 : Résultat du ping de PC4 vers le serveur WEB (REUSSITE)



### 3.6.5 Listing des archives des configurations

Pour lister toutes les configurations archivées dans GASC, l'administrateur clique sur le menu déroulant "Sauver des configs", puis sur le sous lien "Liste des archives", la figure 33 s'ouvre :

La légende est la suivante :

- La couleur grise est mise pour les configurations des routeurs
- La couleur jaune est mise pour les configurations des Switchs

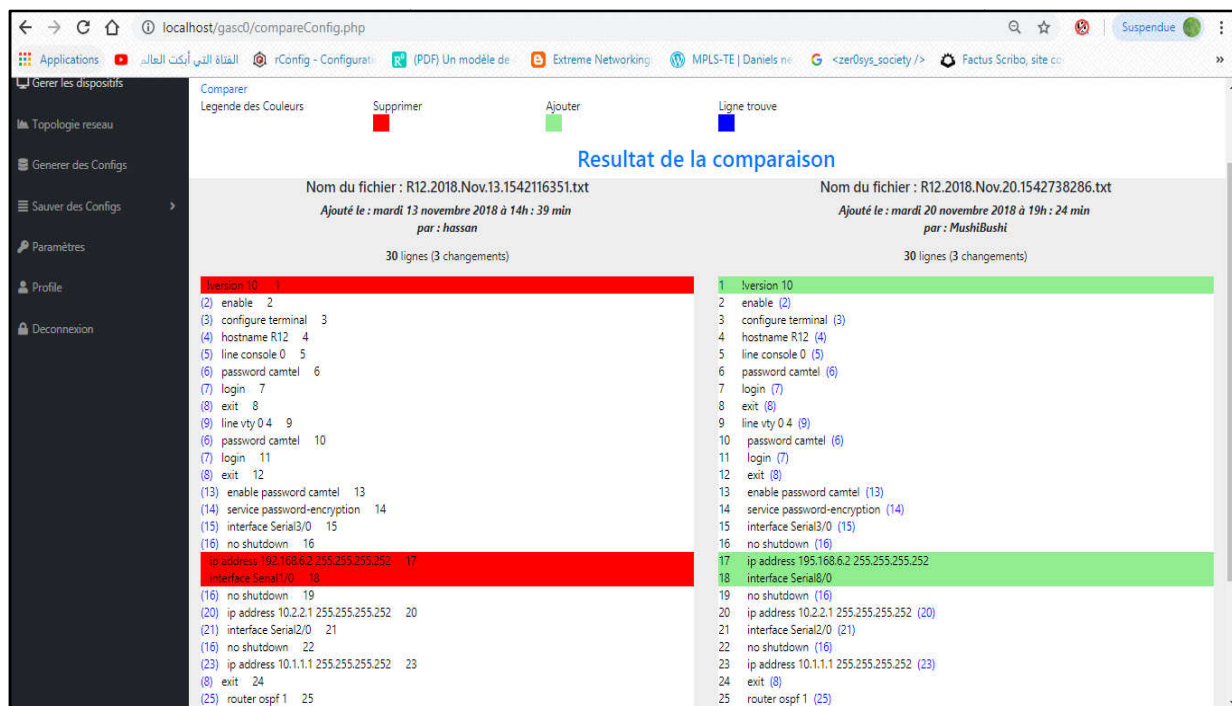
Le tableau contient une zone de recherche fluide qui permet de parcourir les archives.

ID	Noeud	Localisation du fichier	Année	Mois	Jour	Action
1	R11	C:\wamp64\www\gasc0\Configurations/Routeurs/R11/2018/Nov/13/R11.2018.Nov.13.1542116341.txt	2018	novembre	13th	Afficher Telecharger
2	R11	C:\wamp64\www\gasc0\Configurations/Routeurs/R11/2018/Nov/13/R11.2018.Nov.13.1542116349.txt	2018	novembre	13th	Afficher Telecharger
3	R12	C:\wamp64\www\gasc0\Configurations/Routeurs/R12/2018/Nov/13/R12.2018.Nov.13.1542116351.txt	2018	novembre	13th	Afficher Telecharger
4	R13	C:\wamp64\www\gasc0\Configurations/Routeurs/R13/2018/Nov/13/R13.2018.Nov.13.1542116352.txt	2018	novembre	13th	Afficher Telecharger
5	R11	C:\wamp64\www\gasc0\Configurations/Routeurs/R11/2018/Nov/17/R11.2018.Nov.17.1542477816.txt	2018	novembre	17th	Afficher Telecharger
6	R12	C:\wamp64\www\gasc0\Configurations/Routeurs/R12/2018/Nov/17/R12.2018.Nov.17.1542477818.txt	2018	novembre	17th	Afficher Telecharger
7	S11	C:\wamp64\www\gasc0\Configurations/Switchs/S11/2018/Nov/20.1542720007.txt	2018	novembre	20th	Afficher Telecharger
8	S12	C:\wamp64\www\gasc0\Configurations/Switchs/S12/2018/Nov.20.1542720024.txt	2018	novembre	20th	Afficher Telecharger
9	S13	C:\wamp64\www\gasc0\Configurations/Switchs/S13/2018/Nov.20.1542720028.txt	2018	novembre	20th	Afficher Telecharger

Figure 33 : Listing des archives des configurations

### 3.6.6 Comparaison des fichiers des configurations

Pour comparer les fichiers des configurations dans GASC, l'administrateur clique sur le menu déroulant "Sauver des configs", puis sur le sous lien "Comparer des archives", un formulaire de comparaison s'affiche puis l'administrateur choisit les nœuds et leurs différentes archives, puis clique sur le bouton Comparer. Le résultat de la comparaison se trouve dans la figure 34.



**Figure 34 : Résultat de la comparaison des fichiers**

Les différences entre les archives sont mis en évidence suivant la légende des couleurs :

- **Rouge** : pour les lignes supprimées
- **Vert** : pour les lignes ajoutées

Dans ce chapitre, il était question d'affiner ce que nous avons vu tout au long du mémoire et simplifier le travail en des interfaces qui résument l'application de gestion des configurations, nous avons également présenté comment certaines fonctionnalités de l'application sont réalisées et la façon dont GASC sera déployée.

## CONCLUSION GENERALE ET PERSPECTIVES

La gestion des réseaux est un domaine vaste et complexe, quelle que soit la technologie à laquelle elle s'applique. Les configurations réseaux constituent une source d'informations précieuse pour les opérateurs. Devenue incontournable dans le bon fonctionnement d'un réseau, celles-ci requièrent une attention particulière en matière de gestion. GASC a été conçue dans le but de mieux gérer ces configurations, allant de la génération à la sauvegarde de ces dernières.

Les objectifs présentés au début de ce travail ont été atteints : l'application développée (*GASC*) est *flexible* et permet la *génération* de configurations de réseau ainsi que la sauvegarde de ces dernières. L'avantage de générer automatiquement les configurations garantit que la configuration ne contient pas d'erreurs, et ça évite également les oublis de certains paramètres. Notre application est capable également de gérer les sauvegardes car elles sont indispensables pour préserver les configurations en cas de désastre. L'archivage est également pris en compte dans GASC, car il permet de remonter beaucoup plus loin dans le temps afin de faire des enquêtes ou de récupérer des fichiers perdus dans un passé lointain.

Pour arriver à ces résultats, nous avons divisé le présent document en chapitres. Tout d'abord le premier chapitre a été consacré au contexte et à la problématique, tout en fixant les objectifs et la méthodologie de travail adoptée, ainsi que la solution proposée après l'étude et les critiques du système existant.

Le deuxième chapitre a fait l'objet de la méthodologie de conception. Dans un premier temps, nous avons fait un état de l'art en matière de gestion des configurations réseaux, en livrant quelques solutions standardisées par des organismes tels que l'IETF et l'ISO ainsi que la présentation de quelques travaux de recherches universitaires, tout en dégagant les différentes comparaisons entre ces travaux et le nôtre. Dans un second temps, nous avons entamé la conception de la solution en montrant les différents algorithmes utilisés par notre solution. Nous avons également présenté la structure modulaire et l'architecture fonctionnelle de la solution. Nous avons terminé le chapitre par la présentation des différents modèles UML.

Le troisième et dernier chapitre est consacré à la réalisation de l'application, en présentant les différents outils et langage de programmations utilisés pour produire l'application. Il est terminé par la présentation des différents scénarios de l'application sous forme de capture d'écran.

Afin de rendre notre travail plus original, nous proposons dans un avenir proche d'intégrer une approche « **du bas vers le haut** » partant d'une configuration existante dans le but d'automatiser le processus de configuration. Le travail devra alors déboucher sur l'implémentation de ces trois étapes :

1. Un travail de *reverse-engineering* à partir des configurations existantes, afin de reconstruire la topologie du réseau et de résumer son état. Cette phase permet généralement de découvrir des erreurs corrigibles immédiatement par les administrateurs.
2. Une phase de *data Mining* afin d'identifier les politiques locales au réseau choisies par les administrateurs ainsi que les violations détectées par rapport à ces politiques. Cette phase permet de les alerter des inconsistances éventuellement présentes au sein de leur réseau.
3. Au terme de la seconde phase, le réseau dispose d'une configuration « correcte » par rapport à un ensemble de règles. En d'autres termes, il semble avoir été configuré de façon automatique. La troisième étape consiste donc à sauvegarder l'état du réseau dans une base de données afin de faciliter les configurations ultérieures

Nous proposons également de définir, une interface graphique sur laquelle un opérateur pourra dessiner sa topologie réseau, et ensuite obtenir automatiquement les configurations de topologie. Comme notre application ne permet pas la simulation, nous proposons de générer des fichiers spécifiques pouvant être utilisés dans des logiciels de simulation.

En guise de conclusion, nous espérons que ces approches de gestion des configurations réseau sont promises à un bel avenir. En effet, les apports sont réels et les bénéfices nombreux. *GASC* répond déjà à quelques objectifs clés mais ne constitue qu'un premier pas prometteur dans le domaine de la gestion des configuration.

## BIBLIOGRAPHIE

- [1] **Ronan KERYELL** : *Utilisation du logiciel de sauvegarde libre AMANDA*, <http://2001.jres.org/actes/amanda.pdf> consulté le 02 septembre 2018
- [2] **Deca, R., O. Cherkaoui, Y. Savaria, et D. Slone. 2007.** « *Constraint-based model for network service provisioning* », *Annals of telecommunications*, p. 847–870.
- [3] **Campi, N. et K. Bauer. 2009** : *Introducing the Basics of Automation*. Apress, <http://docs.linuxtone.org/ebooks/autOps/%A1%B6LinuxUnix%B2%D9%D7%F7%CF%B5%CD%B3%D7%D4%B6%AF%BB%AF%B9%DC%C0%ED%A1%B7%A3%A8%B5%DA%B6%FE%B0%E6%A3%A9.pdf> consulté le 03 octobre 2018
- [4] **Juniper Networks. 2008.** « *What's behind network downtime ?* », *White Paper*, p. 3–10
- [5] **ÉRIC LUNAUD NGOUPÉ** : *Gestion automatique des configurations réseaux: une approche déductive*, thèse présentée à l'université du Québec à Chicoutimi, 2015.
- [6] **Raj Jain** *Chapter 30 Network Management (SNMP)*, [https://www.cse.wustl.edu/~jain/cis678-97/ftp/f30\\_snm.pdf](https://www.cse.wustl.edu/~jain/cis678-97/ftp/f30_snm.pdf) consulté le 07 Août 2018
- [7] **Miroslav Matuska** : *Metaconfiguration of the Computer Network*, CESNET, Tech. Rep. 27/2004.
- [8] **Khalid El-Arini et Kevin Killourhy** : Bayesian Detection of Router Configuration Anomalies. *In MineNet '05: Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, pages 221–222, New York, NY, USA, 2005. ACM.
- [9] **Laurent VANBEVER et Grégory PARDOEN** : *Design et implémentation d'un logiciel de validation et de génération de configurations réseaux*. thèse présentée à l'université du Québec à Chicoutimi, 2006.
- [10] **Carina Roels** : *Débutez l'analyse logicielle avec UML* : <https://openclassrooms.com/fr/courses/2035826-debutez-lanalyse-logicielle-avec-uml>, consulté en Août
- [11] **Joseph Gabay David Gabay** : *UML 2 , Analyse et conception*, <https://www.eyrolles.com/Informatique/Livre/uml-2-analyse-et-conception-9782100518302/>

- [12] **Ram-0000** : *Introduction à la cryptographie*,  
<https://ram.0000.developpez.com/tutoriels/cryptographie/>, consulté en Novembre
- [13] <https://fr.wikipedia.org/wiki/WampServer>, consulté en Novembre
- [14] **Pierre Christophe** : *Sublime Text 3 et quelques plugins*, <https://www.pccindow.com/fr/sublime-text3-ubuntu-16-04/>, consulté en Novembre