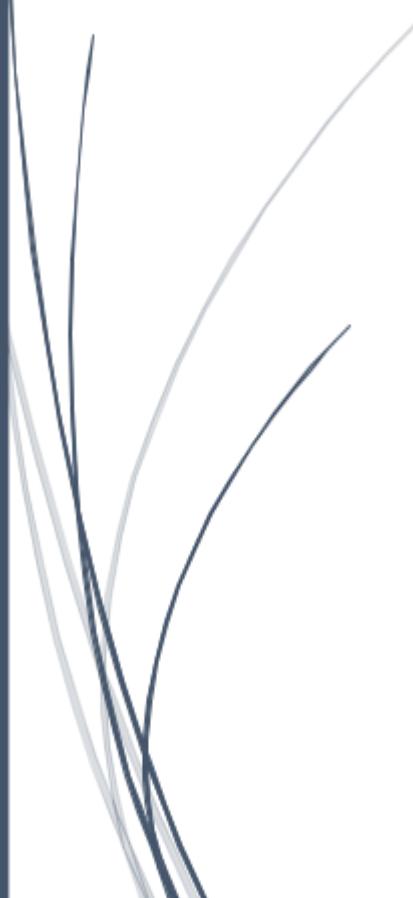


# **DEVELOPPER DES APPLICATIONS COMMUNICANTES**



Réalisé par Baye Niass GUEYE

Etudiant BUT RT21 - IUT Colmar

Devcloud FI

## **Sommaire**

- I. Présentation du projet**
- II. Répartition du travail**
- III. Fonctionnalités prévues**
- IV. Planning de développement**
- V. Organisation et outils**
- VI. Risques identifiés**

## **Présentation du Projet**

Ce projet consiste à créer une application capable de communiquer à travers un réseau de façon sécurisée et anonyme. L'objectif est de mettre en place un système inspiré du principe du “routage en oignon”, où les messages passent par plusieurs routeurs virtuels avant d’arriver au destinataire. Chaque routeur ne connaît qu'une seule information : le prochain routeur vers lequel transmettre les données. Cela permet de garder l'origine du message totalement cachée.

En développant ce projet, j'apprends à utiliser les sockets réseau, le multithreading, le chiffrement et une architecture distribuée.

## **Répartition du travail**

Vu que je réalise ce projet seul, j'ai organisé mon travail comme suit: j'ai commencé par découper le projet en grands blocs : le réseau, le routage, le chiffrement, la base de données, les interfaces et les tests. Ensuite, j'ai défini l'ordre logique pour avancer : d'abord établir la communication réseau, puis le routage, le chiffrement, et enfin les interfaces.

Voici une liste des différentes tâches que je dois réaliser:

- Analyse du cahier des charges
- Création et organisation dépôt Git, Gantt, rapport de gestion de projet
- Développement Master (socket + DB)
- Développement Routeur (multithread + forwarding simple)
- Développement Client minimal (connexion + envoi)
- Implémentation de l'algorithme de chiffrement pédagogique
- Intégration complète (client → routeurs → client final)
- Création des interfaces graphiques PyQt (client + master)
- Tests multi-machines + corrections

Cette organisation me permettra de travailler étape par étape, sans me disperser, tout en gardant une vision claire de l'ensemble du projet.

## **Fonctionnalités prévues**

Le projet doit intégrer plusieurs fonctionnalités essentielles :

- **Routage multi-sauts** : le message passe par plusieurs routeurs avant d’arriver au destinataire.
- **Chiffrement en couches** : le message est protégé par plusieurs couches de chiffrement, comme un oignon.
- **Master du réseau** : un serveur qui gère les routeurs, leurs clés et la topologie.
- **Routeurs virtuels** : chaque routeur reçoit un message, retire une couche de chiffrement et l’envoie au prochain.
- **Client graphique** : une interface simple permettant d’envoyer un message anonymement.
- **Base de données** : stockage des clés, des routes et de certaines informations utiles.  
Ces fonctionnalités seront développées progressivement pour construire un système complet et fonctionnel.

## Planning de développement

Pour bien réaliser ce projet qui dure **65.75h**, j’ai fait un planning de développement sur Notion qui m’a permis de répartir les tâches à faire dans le temps.

Je dois aussi faire un rapport personnel de l’ensemble du projet en le mettant à jour au fur et à mesure avec des tests de fonctionnement

Ci-dessous le lien qui vous montre mon diagramme de GANTT réalisé sur Notion.

[https://www.notion.so/2ae1b049502e8053ad16d49c3e2c6257?v=2af1b049502e804f8d6e000c103214f0&source=copy\\_link](https://www.notion.so/2ae1b049502e8053ad16d49c3e2c6257?v=2af1b049502e804f8d6e000c103214f0&source=copy_link)

## Organisation et outils

Pour organiser mon travail, j’ai utilisé différents outils adaptés au développement :

- **GitHub**: pour y déposer mon travail et faire des push régulièrement pour que le prof puisse suivre mon avancement
- **Notion**: pour faire un tableau de suivi de l’ensemble des tâches .
- **Python**: pour développer le système réseau, les routeurs, le master et les clients.
- **MariaDB**: pour stocker les clés et les routes.
- **Qt**: pour créer les interfaces graphiques.

J’ai également utilisé des machines virtuelles pour tester la communication.

## Risques identifiés

Plusieurs risques peuvent ralentir le projet :

- **La difficulté du chiffrement**, qui peut prendre du temps à mettre en place correctement.
- **Les problèmes de communication entre les routeurs**, surtout lorsqu'il y a plusieurs sauts.
- **Le manque de temps**, car certaines parties comme l'interface Qt sont plus longues que prévu.
- **Les erreurs de configuration de la base de données**, qui peuvent bloquer la progression.

Pour réduire ces risques, je prévois de tester chaque module séparément, d'avancer par petites étapes, et de garder une marge de sécurité dans le planning.