

# Rapport du projet

## **Equipe : Thomas, Rafael, Raphael**

### **1. Organisation et répartition du travail**

Ensemble, dans le cadre d'un projet intitulé C-Wildwater, nous avons travaillé sur le codage d'un programme (en langage C avec un script Shell) permettant de faire la synthèse de données d'un système de distribution d'eau. Chaque membre de l'équipe a contribué activement à différentes étapes du projet, en fonction de ses compétences et disponibilités.

Pour mener à bien ce projet, nous avons mis en place plusieurs réunions à distance tout au long de l'avancement. Ces rendez-vous nous ont permis de :

- Faire le point régulièrement sur le travail accompli,
- Définir les tâches prioritaires pour chacun,
- Identifier les problèmes techniques ou organisationnels rencontrés,
- Répartir les responsabilités de manière équitable.

Concernant la répartition des tâches, Raphael a travaillé sur le code de la partie concernant les avl et leur gestion ainsi que la rédaction de ce rapport et du readme tandis que Rafael s'est occupé de la partie des fuites ainsi que de l'histogramme, enfin Thomas a travaillé sur le makefile ainsi que le « main.c », le « type.h » et à aider Rafael sur certaines fonctions de ses parties du travail et tous les trois avons fait le Shell ensemble afin de tous connaître cette nouvelle façon de gérer les données. Cette répartition que l'on trouve équitable a été faite en fonctions du niveau et préférences de chacun.

Certaines difficultés ont été rencontrées, notamment par rapport à la répartition du travail durant les premiers jours car nous travaillions tous les 3 sur la même partie de code afin d'être sûr d'être tous les 3 en accord afin de passer aux fonctions suivantes puis nous nous sommes fait confiance et avons décidé de la répartition des tâches comme énoncé précédemment, ce qui était beaucoup plus simple en termes d'organisation et d'avancée.

Enfin, bien qu'il soit encore perfectible (codage du bonus par exemple), ce projet fonctionnel nous a permis de mettre en pratique nos connaissances en programmation, en gestion de projet et en travail collaboratif.

## 2. Analyse des résultats obtenus

- Analyse des performances :

En utilisant des AVL, la recherche d'une usine parmi des milliers de lignes se fait en complexité logarithmique ( $O(\log n)$ ).

On remarque d'ailleurs (dans les exemples\_code\_1 et exemple\_code\_2) que l'exécution du programme prends quelques millisecondes à quelques secondes, ce qui est assez rapide pour ce type de programme.

- Analyse des histogrammes :

Que ce soit pour la v0 ou la v3, on remarque que la capacité maximale est toujours supérieure à celle réelle (voire max.png et real.png). Ce qui montre le bon fonctionnement de notre code. En effet, une usine ne tourne jamais à 100% de sa capacité maximale pour garder une marge de sécurité ou à cause des pertes en amont.

- Analyse des fuites (leaks) :

De même concernant les volumes, que se soit pour la v0 ou la v3, on remarque que le volume maximal est toujours supérieur à celui réel (voire vol\_max.dat et vol\_real.dat). Ce qui montre le bon fonctionnement de notre code. En effet, suite aux fuites, le volume diminue plus ou moins selon le nombre de pertes.