

EISH Ahmed

MOHAMED Naim

ITIM Tarek

Pré-Ing 2

2025-2026



## **Carnet de bord du groupe Pré-Ing 2 du projet CY WildWater**

# **Table des matières :**

Répartition des tâches au sein du groupe : 3

Planning de réalisation : 3-4

Difficultés rencontrées : 5

les limites du programme : 5

## Répartition des tâches au sein du groupe :

Tarek s'est occupé de la structure de données en C (Structures AVL, Noeuds), de la gestion de la mémoire et de l'implémentation des rotations et de l'équilibrage dans fonction.c.

Naim a pris en charge le Script Shell (shell.sh) pour l'automatisation et la génération des graphiques avec Gnuplot, ainsi que le parsing du fichier CSV (split\_5\_colonnes).

Ahmed a travaillé sur la fonctionnalité des fuites (leaks), la gestion du graphe de distribution, le Makefile, et a assuré la coordination globale et la rédaction de ce carnet de bord.

## Planning de réalisation :

### Planning théorique :

#### 1. PHASE 1 : Les Fondations

- Définir les structures de données (Structures C et AVL).
- Coder les fonctions de base de l'AVL (Insertion, Rotation, Équilibrage).
- Test : Créer un main basique pour vérifier l'intégrité de l'arbre.

#### 2. PHASE 2 : Le Parsing (Lecture du CSV)

- Coder la lecture du fichier ligne par ligne.
- Remplir l'AVL avec les données réelles (filtrage pour ne garder que les usines au début).
- Test : Vérifier qu'on a bien toutes les usines en mémoire.

#### 3. PHASE 3 : Traitement Histogramme (Histo)

- Ajouter la logique de sommation (Capacité, Volume Source, Volume Traité).
- Générer le fichier .dat temporaire de sortie.

#### **4. PHASE 4 : Le Script Shell**

- Coder le script qui compile le C, lance le traitement "Histo", et appelle Gnuplot.

#### **5. PHASE 5 : Les Fuites**

- Modifier des structures pour gérer la partie des fuites (graphe orienté pour le réseau aval).

### **En pratique :**

- Création du dépôt git et lecture intensive du sujet. Répartition des rôles.
- Structure du projet + Début d'écriture du code (fichiers .h et .c).
- Écriture des fonctions AVL (rotations, insertions) + début du travail sur le parsing du fichier.
- Création du Shell (shell.sh) et intégration des premières commandes pour compiler via le Makefile.
- En Shell : gestion des arguments et intégration de Gnuplot. En C : finalisation de la lecture du fichier et du remplissage de l'AVL pour les histogrammes.
- Fin du Makefile et validation de la partie Histogrammes (Top 10 / Bottom 50).
- Gros travail sur la fonction leaks. Implémentation d'un index AVL secondaire pour construire le graphe des fuites efficacement. Correction de bugs sur le calcul des volumes.
- Nettoyage du code, ajout des commentaires, rédaction du Readme et de ce carnet de bord.

## **Difficultés rencontrées :**

Tout d'abord, la gestion de la mémoire en C a été un défi majeur. Assurer que toutes les allocations dynamiques (malloc) pour les nœuds des graphes et les structures AVL soient correctement libérées (free) nous a demandé beaucoup de temps de débogage.

Le fait de lier le C, le Shell et Gnuplot ensemble représentait un obstacle, notamment pour formater les données exactement comme Gnuplot l'attendait.

Ensuite, la complexité de l'algorithme pour la partie "Leaks" n'était pas évidente au premier abord. Il a fallu comprendre comment reconstruire un graphe orienté à partir d'un fichier CSV où les lignes ne sont pas triées, ce qui nous a obligés à utiliser une structure d'indexation supplémentaire pour ne pas exploser les temps de calcul.

De plus, notre gestion du temps a été mise à l'épreuve par la période d'examens et les autres projets en parallèle, ce qui a légèrement décalé notre planning théorique.

Enfin, pour la commande "leaks", nous avons eu des difficultés à interpréter correctement les volumes : nous confondions parfois la capacité maximale théorique de l'usine avec le volume réel entrant, ce qui faussait les résultats de pertes.

## **Limites du projet :**

Toutes les fonctionnalités de base sont implémentées (histo et leaks).

Une limite concerne la performance sur des machines très modestes pour la commande leaks sur de très grosses usines, car la construction du graphe consomme de la mémoire. De plus, le script Shell dépend de la présence de Gnuplot sur la machine hôte.