



C-WILDWATER

PréING2 MI2-G



AUTEURS

SARAH XERRI

BELQUASS HAJAR

RENOUA FASSI FEHRI

CY Tech

2025/2026

Rapport de projet – WildWater

Pour ce projet d'informatique, nous avons travaillé en équipe afin de développer une application combinant un script Shell et un programme en langage C. La répartition des tâches au sein du groupe nous a permis d'avancer de manière organisée et efficace tout au long du projet. Afin de faciliter la communication et le partage des informations, nous avons utilisé un dépôt GitHub commun, ce qui nous a permis de collaborer efficacement et de gérer les différentes versions du code.

Sarah s'est occupée du développement du script Shell. Elle a également pris en charge la rédaction du fichier README, dans lequel sont expliquées les instructions nécessaires pour compiler et utiliser correctement l'application. Ce fichier permet à tout utilisateur de comprendre rapidement le fonctionnement du programme et la manière de l'exécuter.

De mon côté, Hajar, je me suis également occupée du développement du script Shell. J'ai en outre pris en charge la création et la configuration du Makefile, utilisé pour compiler automatiquement le programme en langage C et générer l'exécutable nécessaire au bon fonctionnement de l'application.

Renoua s'est chargé du développement du programme en langage C, qui est responsable de l'ensemble des traitements demandés dans le projet. Ce programme lit le fichier de données fourni en entrée et réalise les calculs nécessaires, tels que la génération des histogrammes des usines ou le calcul des volumes de fuites, avant de produire les fichiers de sortie correspondants.

L'application est organisée autour de plusieurs fichiers principaux. Le script Shell constitue le point d'entrée du projet et permet de séquencer toutes les actions demandées par l'utilisateur. Le programme en langage C regroupe l'ensemble des traitements et calculs imposés par le cahier des charges. Le Makefile permet de compiler automatiquement le programme C, garantissant ainsi une exécution correcte de l'application. Enfin, le fichier README fournit les instructions nécessaires à la compilation et à l'utilisation du projet. Notre programme a été conçu de manière générique, de façon à fonctionner avec tout fichier de données respectant la même structure que celui fourni initialement.

Lors de la réalisation de ce projet, nous avons rencontré certaines difficultés, principalement liées à la gestion du temps et à la coordination entre les différentes parties du programme. Le travail sur le script Shell a nécessité une attention particulière afin de respecter strictement les consignes du sujet et d'assurer une bonne communication avec le programme en langage C. De plus, lors de l'exécution de l'application sur le fichier de données volumineux, nous avons observé que le calcul des rendements (ainsi que les tris associés) nécessite un temps de traitement assez long (environ 20 secondes voir quelques minutes). Cette latence s'explique par la complexité algorithmique du traitement de millions de lignes en langage C, notamment lors des étapes de lecture, de stockage en mémoire et de tri final avant la génération des résultats. Bien que le programme soit optimisé, la gestion de fichiers de cette envergure souligne l'importance de l'efficacité des structures de données utilisées. Toutefois, ces contraintes nous ont

permis de mieux nous organiser, de renforcer notre collaboration et d'améliorer la qualité globale du projet.

En analysant les résultats mathématiques et graphiques générés par le programme on peut en conclure qu'il existe une forte disparité entre les installations. On identifie d'un côté des complexes majeurs traitant des volumes extrêmement élevés, et de l'autre, une multitude de petites unités aux débits plus modestes. De plus, on remarque que certaines petites usines peuvent subir des pics de charge très importants, dépassant largement leur volume d'activité habituel. Cela nous montre donc que la plupart des pertes d'eau viennent de quelques grandes installations mal optimisées, tandis que les petites usines restent plus vulnérables aux variations de débit. Ces observations permettent de cibler les réparations là où elles auront le plus d'impact, à la fois pour économiser l'eau et pour alléger les coûts.

Au-delà de ces conclusions opérationnelles sur la gestion du réseau, la mise en œuvre technique de ces analyses a constitué un véritable levier d'apprentissage pour notre équipe.

Ce projet nous a permis de renforcer nos compétences en programmation, aussi bien en langage C qu'en Shell. Il nous a également permis de découvrir l'importance de l'automatisation à l'aide de scripts Shell et de la gestion de projets via GitHub. Ces compétences acquises seront utiles pour nos futurs projets académiques et professionnels.