



# Rapport de Projet : Synthèse des données de distribution d'électricité

## Introduction

Ce projet avait pour objectif la création d'un programme permettant de synthétiser les données d'un vaste réseau de distribution d'électricité. Nous devions traiter un fichier CSV volumineux contenant les informations sur les centrales électriques, les stations HV-B (haute tension), HV-A (tension intermédiaire), les postes LV (basse tension), ainsi que les consommateurs finaux (particuliers et entreprises). L'objectif était de déterminer les états de surproduction ou sous-production des stations et d'analyser la consommation.

Le projet a été réalisé par une équipe de trois membres : Chady, Aline et Léo. Ce rapport détaille la répartition des tâches, le planning, les difficultés rencontrées, les solutions apportées, ainsi que notre retour critique et des propositions d'amélioration.

---

## Répartition des tâches

### Aline : Script Shell

Aline était responsable de la conception et de la mise en place du script Shell. Ses principales contributions incluent :

- **Validation des paramètres** : Gestion des arguments en ligne de commande, vérification de la présence et de la cohérence des options obligatoires (« station », « consommateur », fichier).
- **Gestion des dossiers** : Création des dossiers « tmp » et « graphs », vidage des anciens fichiers temporaires.
- **Extraction des données** : Utilisation de commandes Shell (« grep », « awk ») pour filtrer les informations pertinentes à transmettre au programme C.
- **Intégration avec le programme C** : Compilation automatique (« make ») et exécution de l'exécutable « cwire ».
- **Gestion des erreurs** : Affichage des messages d'aide et des états d'erreur avec des explications claires.

### Chady : Programme C

Chady était en charge du développement du programme principal en C, avec un accent sur l'optimisation et la gestion des données :

- **Implémentation de l'AVL** : Conception d'un arbre AVL (auto-équilibré) pour optimiser la recherche et l'agrégation des données.
- **Traitement des stations** : Ajout et mise à jour des stations HV-B, HV-A et LV avec leurs capacités et consommations.
- **Gestion des consommateurs** : Calcul des sommes de consommation pour les entreprises et particuliers selon les besoins.
- **Robustesse** : Gestion des erreurs en renvoyant des codes d'état clairs et en assurant la libération mémoire.
- **Optimisation** : Minimisation de l'empreinte mémoire en structurant les données efficacement.

## Léo : Coordination et Documentation

Léo était chargé de la coordination globale et de la documentation du projet :

- **Planification et suivi** : Organisation des réunions hebdomadaires, suivi des étapes de développement.
  - **Tests** : Conception de scénarios de test pour valider l'intégration entre le Shell et le C.
  - **Documentation** : Rédaction du fichier ReadMe, préparation des instructions pour compiler et exécuter le projet.
  - **Livraison** : Gestion du dépôt GitHub, incluant l'organisation des fichiers et l'accessibilité publique.
- 

## Planning semaine par semaine

### Semaine 1 : Analyse et initialisation

- **Objectifs** : Comprendre les spécifications et organiser le projet.
- **Activités** :
  - Lecture et analyse des requis (ensemble).
  - Définition de la répartition des tâches.
  - Exploration du fichier CSV pour comprendre sa structure et ses particularités.
  - Mise en place de l'arborescence du projet (« input », « codeC », « tmp », « graphs »).

### Semaine 2 : Développement

- **Objectifs** : Commencer le codage des éléments clés.
- **Activités** :
  - Aline : Conception du script Shell, écriture des fonctions pour valider les arguments et extraire les données avec « grep ».
  - Chady : Implémentation de l'AVL et des fonctions associées (insertion, recherche, équilibrage).
  - Léo : Tests unitaires pour vérifier les données extraites et le comportement de l'AVL avec des exemples simples.

## Semaine 3 : Intégration et finalisation

- **Objectifs** : Finaliser le projet et livrer un produit fonctionnel.
  - **Activités** :
    - Aline : Intégration du Shell et du programme C, gestion des fichiers temporaires.
    - Chady : Optimisation des performances, gestion des erreurs dans le programme C.
    - Léo : Finalisation de la documentation et publication sur GitHub.
    - Tous : Validation des scénarios de test et création des fichiers de sortie exigés.
- 

## Difficultés rencontrées et solutions

### Traitement des données volumineuses

- **Problème** : Le fichier CSV contient plus de 5 millions de lignes et est trop volumineux pour être manipulé directement.
- **Solution** : Utilisation de commandes Shell comme « grep » et « awk » pour filtrer les données nécessaires avant de les transmettre au programme C.

### Optimisation des performances

- **Problème** : Les calculs des sommes pour les postes LV étaient trop lents avec de grandes quantités de consommateurs.
- **Solution** : L'utilisation de l'AVL a permis de réduire le temps de traitement grâce à son équilibrage automatique.

### Problèmes d'intégration

- **Problème** : Les fichiers temporairement créés pour l'échange entre le script Shell et le programme C étaient parfois mal formatés.
- **Solution** : Normalisation des fichiers échangés (« tmp/temp\_station.txt » et « tmp/temp\_usager.txt ») avec des délimiteurs cohérents.

### Gestion des erreurs

- **Problème** : Arrêts inattendus du programme C en cas de données invalides.
  - **Solution** : Ajout d'une gestion des exceptions pour détecter les entrées non conformes et renvoyer un code d'erreur approprié.
- 

## Finalisation du projet

### Objectifs atteints

Le projet a répondu à toutes les exigences, notamment :

- La validation des paramètres en ligne de commande.
- Le traitement des différents types de stations (HV-B, HV-A, LV).
- La distinction entre les consommateurs particuliers et entreprises.

- La génération des fichiers de sortie correctement triés et structurés.
- Le respect des contraintes de performance.

## Bonus réalisé

Un graphique a été créé avec GnuPlot pour représenter les 10 postes LV les plus chargés et les 10 moins chargés. Les couleurs permettent une visualisation claire des excès ou marges d'énergie.

## Livraison

Le projet a été publié sur un dépôt GitHub public avec :

- Le code source complet.
  - Les fichiers de test et résultats.
  - Un fichier ReadMe expliquant la compilation et l'utilisation.
- 

## Retour critique

### Points positifs

- Une répartition des tâches claire et efficace.
- Une communication fluide entre les membres de l'équipe.
- Une robustesse du programme grâce à l'AVL et aux validations des entrées.
- Respect des délais impartis.

### Points d'amélioration

- Tester davantage avec des données réelles pour anticiper d'éventuelles anomalies.
  - Intégrer une interface utilisateur pour simplifier l'utilisation.
  - Automatiser la génération des rapports graphiques et statistiques.
- 

## Propositions d'ouverture

### Améliorations fonctionnelles

- **Analyse des tendances** : Intégrer une fonctionnalité pour suivre l'évolution des consommations au fil du temps.
- **Alertes en temps réel** : Ajouter un système de détection des surcharges ou coupures imminentes.

### Extensions technologiques

- **Formats de données** : Supporter des formats supplémentaires comme JSON ou SQL pour une intégration plus facile avec d'autres systèmes.
- **Interopérabilité** : Créer une API REST pour permettre des requêtes dynamiques.

## Perspectives écologiques

- **Optimisation énergétique** : Identifier les postes les moins efficaces pour réduire les pertes d'énergie.
- **Intégration des énergies renouvelables** : Adapter le système pour inclure des sources comme le solaire ou l'éolien.

---

Ce projet a été une réussite grâce à une bonne coordination, des choix techniques pertinents et une implication forte de chaque membre. Il constitue une base solide pour des applications futures dans le domaine de la gestion énergétique.