

# RAPPORT PROJET INFORMATIQUE C-WIRE

---

**Contributeurs:**

KELLAI Alicia (P2-MI2)

MESBAHI Hiba (P2-MI2)

OZSOLAK Atahan (P2-MI2)

**Organisation du groupe:**

Conformément aux attendus du projet, nous avons utilisé GitHub comme environnement d'hébergement. La gestion des commits s'est avérée être une fonctionnalité clé, permettant à chacun de suivre en temps réel les modifications apportées par les autres membres du groupe. Un groupe WhatsApp a été créé pour communiquer. Hiba et Alicia travaillaient sur iOS, tandis que Atahan travaillait sur Windows. Nous nous réunissions régulièrement pour mettre en commun nos avancées et résoudre les éventuels problèmes de compatibilité. En phase finale, Atahan a installé une machine virtuelle Linux pour uniformiser l'environnement de développement et faciliter les tests finaux.

**Répartition des tâches:**

**Hiba** : a conçu la structure initiale de la partie Shell, réalisé la partie C, et rédigé le Makefile.

**Alicia** : a développé en profondeur la partie Shell, écrit les commentaires et vérifié la cohérence du C.

**Atahan** : a travaillé sur le gnuplot, testé le projet et effectué la révision finale de celui-ci.

**Planning de réalisation:**

<b>semaine/ équipe</b>	<b>Hiba</b>	<b>Alicia</b>	<b>Atahan</b>
<b>Semaine 1 (25/11)</b>	Lecture et compréhension du projet, définition des tâches initiales	Lecture et compréhension du projet, définition des tâches initiales	Lecture et compréhension du projet, définition des tâches initiales
<b>Semaine 2 (02/12)</b>	Conception de la structure initiale de la partie Shell, début de la partie C (AVL, fichiers.c et .h)	Début de l'implémentation de la partie Shell	Conception de la structure initiale de la partie C (AVL)
<b>Semaine 3 (09/12)</b>	Finalisation de la partie C, rédaction du Makefile	Développement en profondeur de la partie Shell, correction de la partie C	Rédaction du README et correction du Makefile
<b>Semaine 4 (16/12)</b>	Réécriture de certaines fonctions initialement en C en Shell	réécriture de certaines fonctions initialement en C en Shell, rédaction des commentaires	Tests unitaires sur les parties développées et mise en place de gnuplot

### **Difficultés rencontrées:**

L'un des défis majeurs de notre projet a été la gestion des environnements de développement distincts. En effet, Hiba et Alicia codaient sur iOS, tandis qu'Atahan travaillait sous Windows, nous avons rencontré des difficultés pour synchroniser le travail de chacun. Pour exécuter l'ensemble du projet, un environnement Linux était nécessaire en raison de l'intégration du Makefile, essentiel pour la compilation et l'exécution des différents modules. Cela a impliqué la mise en place d'une machine virtuelle Linux sur la machine d'Atahan.

Concernant le développement, Hiba, par manque de connaissance approfondie en Shell, a initialement implémenté certaines fonctions liées au passage des données à l'AVL, au tri par capacité croissante, ainsi que l'écriture des fichiers de sortie (tels que "lv\_min\_max.csv") en C. Cependant,

après discussion avec Alicia, il a été convenu que ces opérations seraient plus efficaces si elles étaient traitées directement en Shell. Cette décision a permis d'optimiser le traitement et de rendre le code plus cohérent avec l'architecture globale du projet.

Un autre obstacle notable a été le temps de traitement des dossiers. Lors de nos premiers tests, le traitement des données prenait plus d'une minute, ce qui était trop long pour garantir une efficacité optimale. Cela a nécessité une réévaluation de l'approche de traitement des données et des ajustements dans l'optimisation des algorithmes utilisés.

### **Limitations Fonctionnelles:**

Actuellement, le programme ne prend en charge que les fichiers au format `.dat` pour l'exécution. Si l'utilisateur tente de fournir un fichier `.csv` en argument, le programme ne fonctionnera pas correctement, car il n'a pas été conçu pour traiter ce type de fichier.