

Projet: Utilitaire d'Analyse de Fichiers Texte

Introduction

Une entreprise souhaite mettre en place une application en ligne de commande (CLI) pour permettre à ses employés de réaliser des analyses simples de fichiers telles que compter le nombre de lignes, compter le nombre de mots, compter le nombre de caractères et le transformer en majuscule. L'ancienne méthode consistait à utiliser ouvrir les fichiers à l'aide d'un éditeur de texte et effectuer ces opérations manuellement, ce qui posait plusieurs problèmes :

- Difficulté à ouvrir certains fichiers à cause de l'encodage
- Erreurs humaines de saisie des opérations
- Les résultats n'étaient pas consistants en fonction de l'éditeur de texte utilisé

Objectif du Projet

Le but de ce projet est de développer un **outil en ligne de commande (CLI)** en Python qui peut lire et effectuer des analyses statistiques simples sur des fichiers texte.

Les utilisateurs de l'application seront les employés de l'entreprise. Ils pourront effectuer des opérations directement depuis leur terminal. Comme ils utilisent plusieurs systèmes d'exploitation, l'application sera disponible dans une image docker.

Afin de s'assurer que l'application correspond bien aux utilisateurs, il sera demandé de faire au moins 2 sprints scrums pour tester les fonctionnalités.

Fonctionnalités

L'application doit proposer les quatre fonctionnalités suivantes, prenant toutes un chemin de fichier texte (.txt) comme argument principal :

Fonctionnalité	Description	Exemple de commande
lines	Compter et afficher le nombre total de lignes dans le fichier.	python file-analyzer.py lines data.txt

words	Compter et afficher le nombre total de mots (séparés par des espaces) dans le fichier.	python file-analyzer.py words data.txt
char	Compter et afficher le nombre d'occurrences d'un caractère spécifique , fourni comme second argument.	python file-analyzer.py char data.txt a
upper	Lire le contenu complet du fichier et l'afficher dans le terminal, entièrement en majuscules .	python file-analyzer.py upper data.txt

Gestion des Erreurs

L'application doit gérer les cas d'erreur suivants :

1. **Fichier non trouvé** : Si le chemin de fichier fourni n'existe pas ou est incorrect.
2. **Argument manquant** : Si l'utilisateur exécute la commande char sans fournir le caractère à rechercher.

Livrables et Contraintes

Les livrables attendus devront fournir les éléments suivants, qui sont cruciaux pour l'évaluation :

1. **Un cahier des charges détaillé**
2. **Un dépôt Gitlab** contenant le code source de l'application, avec les commits détaillés, les branches et les tags associées ainsi que l'intégration continue
3. **Un fichier Dockerfile** permettant de containeriser l'application
4. **Un fichier d'intégration continue GitLab CI** permettant d'exécuter les tests du projet, de générer la documentation et de créer le conteneur applicatif.

Conclusion

Ce projet vise à moderniser et à améliorer l'efficacité des analyses de fichiers au sein de l'entreprise en remplaçant les opérations manuelles par une application en ligne de commande.

Cette solution permettra non seulement de réduire les erreurs humaines, mais aussi de garantir une accessibilité et une utilisation plus rapides, standardisée et intuitives pour tous les employés.