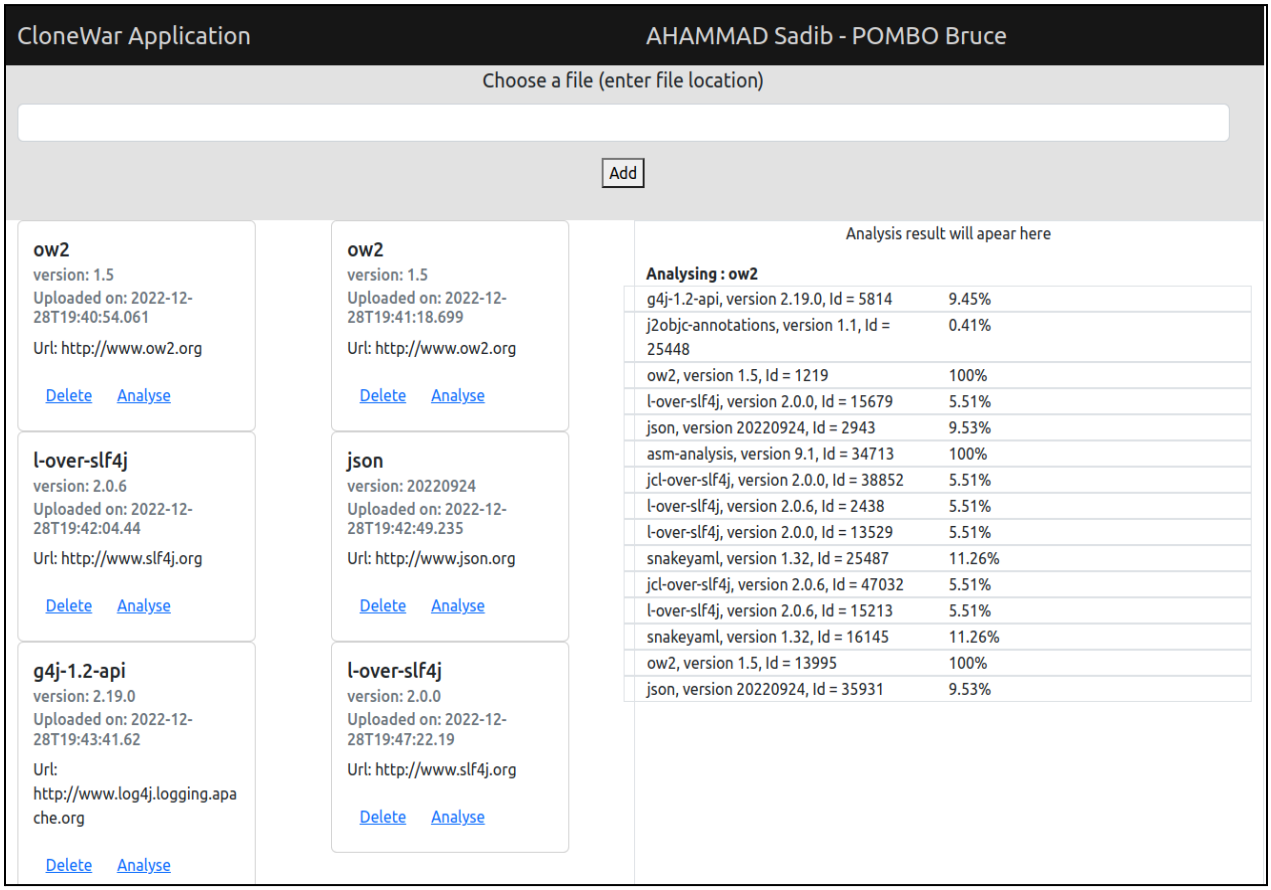
Projet Clonewar

M1 INFORMATIQUE - IGM

Binôme: AHAMMAD Sadib - POMBO Bruce



Université Gustave Eiffel

INTRODUCTION

Nous avons réalisé ce projet de détection de clones en utilisant Java comme langage de programmation, Spring Boot comme serveur, SQLite comme base de données, Vue.js et Bootstrap en front-end.

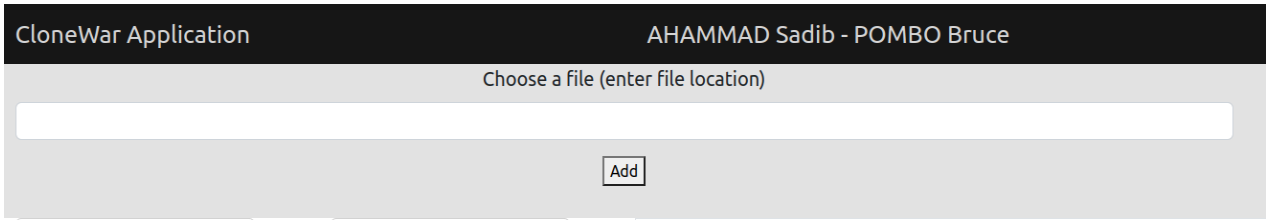
Le projet consistait à développer une application permettant de détecter les clones dans un archive en jar en utilisant l'algorithme de Karp-Rabin. L'application permet à l'utilisateur d'ajouter un artefact et de la comparer avec celles présentes dans la base de données. Si un code similaire est trouvée, l'application affiche tous les artefacts avec le pourcentage de code cloné.

En ce qui concerne la répartition des tâches, nous avons travaillé en binôme pour réaliser ce projet. L'un de nous s'est occupé de la mise en place du serveur Spring Boot et de la gestion de la base de données SQLite, tandis que l'autre a développé l'interface utilisateur en utilisant Vue.js et Bootstrap. Nous avons également travaillé ensemble pour écrire le code Java permettant la comparaison des codes.

UTILISATION

Pour lancer l'application en mode de développement, voici les étapes suivantes:

1. Ouvrez IntelliJ et ouvrez le projet Clonewar.
2. Cliquez sur le bouton "Run" ou utilisez le raccourci clavier "Shift + F10" pour lancer l'application.
3. Une fois le serveur Spring Boot lancé, ouvrez un terminal et accédez au dossier "front" du projet.
4. Assurez-vous d'avoir installé Node.js et npm sur votre ordinateur.
5. Dans le terminal, lancez la commande "npm install" pour installer les dépendances du projet.
6. Une fois l'installation terminée, lancez la commande "npm run serve" pour lancer le serveur de développement de Vue.js. Le serveur se lancera sur le port 9000 par défaut.
7. Ouvrez votre navigateur et accédez à l'adresse "http://localhost:9000" pour accéder à l'application.



Pour ajouter un artefact il faut mettre le chemin vers le répertoire où se trouve l'archive Jar sans spécifier le nom de l'archive.

*Exemple: si le fichier **Babylon.jar** se trouve dans le répertoire mes-projets, il faudrait mettre : **/home/user/Desktop/mes-projet/***

DETECTION DE CLONE

1. **Abstraction de bytecode**
   Puisqu'on devait utiliser le bytecode généré pour détecter les clones en utilisant l'API de ClassVisitor, nous avons créé une classe dans notre projet qu'on a appelé MyclassVistor implémentant ClassVisitor pour pouvoir définir quelques méthodes nous permettant d'abstraire le bytecode.

2. **Calcul de Hash**
   Nous avons défini une fenêtre de taille 2 sur le nombre d'instructions essentielles pour calculer le hash de sorte que : soit S une chaîne qui contient les instructions dont on souhaite calculer le hash, ceci se calcule alors en **hash = ΣC(a)*i** telle que a est un caractère présent dans la chaine representant les instructions, C(a) le code ascii de a et i la position de a dans la chaîne S. nous nous sommes rassurés que le calcule ne produira pas de collision car nous avons abstrait le code de sort que deux suites d'instructions différentes ne produisent pas le même hash. Et nous avons testé avec différentes tailles de fenêtre nous avons réalisé qu'une taille de 2 instructions était le meilleur candidat, nous avons donc pris une fenêtre de taille 2.

3. **Détection de clone**
   Une fois calculé, nous stockons le hash, le numéro, dans le fichier, de ligne de la première instruction dans la fenêtre, le nom de l'artefact, l'id de l'artefact dans une base de donnée. Pour détecter les clones, pour une fenêtre d'instructions donnée, nous regardons dans la base de données s'il existe une autre fenêtre d'instruction qui a le même hash et dont l'id de l'artefact est différent. Pour le faire sur chaque autre artefact que celui analysé, on procède comme suit : soit idArtClone l'id de l'artefact potentiel clone de l'artefact analysé, pour chaque fenêtre d'instruction ayant l'id de l'artefact analysé, on vérifie s'il y'a une autre fenêtre d'instruction ayant l'idArtClone avec le même hash, on incrémente un compteur dont on multipliera par 100 et divisera par le nombre des fenêtre d'instructions de l'artefact analysé. Le résultat constitue le pourcentage cloné entre l'artefact analysé et l'artefact d'id = idArtClone.