DB : base de données avant synchronisation, contient la dernière bonne synchro des arbres.

A : fichier contenant toutes les metadatas actuelles de l’arbre 1

B : fichier contenant toutes les metadatas actuelles de l’arbre 2

* Lire dans la DB ligne par ligne :
  + Compare la ligne lue à toutes les lignes de A : Si il y a match, stocke la ligne matchée dans la variable nline1. Sinon, stocke 0 dans la variable.
  + Idem pour B, stockage dans nline3.
    - Si les 2 variables nline1 et nline3 ne sont pas nulles :
      * On supprime les lignes dans A et B contenant les metadatas dudit fichier car la synchronisation est réussie.
    - Si les 2 variables sont nulles, alors le fichier a été soit modifié, soit supprimé :
      * On cherche dans A le **nom** du fichier. On stocke à nouveau dans nline1 la ligne où le fichier se trouve dans A, et si pas de match on stocke 0.
      * Idem pour B, stockage dans nline3.
      * Si les 2 variables sont nulles, alors le fichier a été supprimé :
        + On supprime la metadata dans la DB.
      * Si [ nline1 == 0 ] et [ nline3 != 0 ], alors le fichier a été modifié dans B et supprimé dans A :
        + On copie le fichier de abre2 dans arbre1
        + On met à jour la metadata de la DB.
      * Idem pour le cas de suppression dans B et modification dans A
      * Si les 2 variables != 0, alors il y a conflit, le fichier a été modifié des 2 côtés :
        + Gestion de conflit.
    - Si [ nline3 == 0 ] et [ nline1 != 0 ], alors le fichier a changé dans arbre1 mais pas dans abre2. On veut savoir ce qui lui est arrivé :
      * Même chose qu’au-dessus, on match que par le nom du fichier.
      * Si le nom du fichier existe dans B, alors il est modifié :
        + Copier le fichier depuis arbre2 dans arbre1
        + Mettre à jour le fichier dans DB.
      * S’il n’existe pas dans B, il est supprimé :
        + Supprimer le fichier dans arbre1
        + Supprimer la metadata dans DB.
    - Idem que le cas au-dessus, mais pour A.

WOAW ! On a réussi à synchroniser tous les fichiers qui était présents dans la base de données DB!

Il faut maintenant s’occuper des nouveaux fichiers qui viennent de pop et dont la DB n’a pas encore connaissance. A présent, après synchronisation des anciens fichiers de la DB, on a supprimé toute la metadata des anciens fichiers présents dans A et B . Il ne reste donc que la metadata des nouveaux fichiers qui viennent d’être ajoutés.

* Lire le fichier A, ligne par ligne :
  + Stocke **le nom** du nouveau fichier dans la variable name1
  + Cherche un match avec les **noms** du fichier B. Si match, stocke le numéro de ligne dans nlineS. Sinon, stocke 0.
  + Si [ nlineS == 0 ] , alors le fichier n’existe pas dans arbre2 :
    - Copier le fichier depuis arbre 1 dans arbre 2.
    - Mettre à jour la DB
  + Sinon, alors le fichier existe dans arbre2. Il reste à savoir si ledit fichier est conforme à sa version dans arbre1 ou les metadatas diffèrent :
    - Compare **la ligne** lue sur A à toutes **les lignes** dans B. Si match, stocke le numéro de la ligne matchée dans la variable nlineS. Sinon, stocke 0.
    - Si [ nlineS !=0 ], alors le fichier dont la DB n’a pas connaissance existe en 2 copies conforme dans arbre1 et 2 :
      * Supprime la ligne dans A et B.
      * Mettre à jour la DB.
    - Sinon, alors le fichier existe dans les 2 arbres, mais les mdat diffèrent, il y a conflit :
      * Gestion de conflit.
* Lire le fichier B, idem.

S’il reste des dossiers non traités avec des fichiers non traités dedans