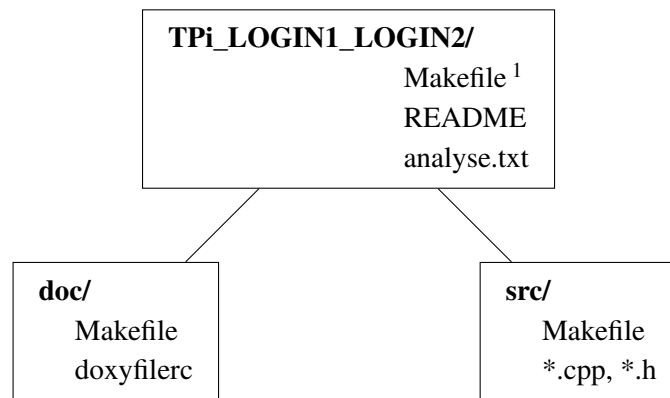


## Instructions pour le rendu des TP

La correction des TP se fait de manière automatique. Il vous faudra donc impérativement respecter les règles ci-dessous pour permettre cette automatisation. Le non-respect de ces règles engendrera une note nulle.

1. Le rapport se fera sous la forme d'une archive (`TPi_LOGIN1_LOGIN2.tar.gz` ou `TPi_LOGIN.tar.gz` si vous êtes seul).
2. **Attention !** Les équipes mixtes IF-MMIS ne sont pas autorisées.
3. Le rapport du  $TP_n$  est à rendre **48H avant** le  $TP_{n+1}$ , pour les TP se déroulant sur deux séances ( $TP_2$  et  $TP_4$ ), aucun rapport intermédiaire n'est demandé.
4. L'arborescence de votre répertoire (contenu de l'archive) doit être :



5. Vérification de votre projet (nécessite Python 2.7 ou supérieur) :  
**Tout au long de votre développement :** Lancez régulièrement le script `verifier.py TPi_LOGIN1_LOGIN2` afin de vérifier que votre dossier est conforme aux instructions données  
**Avant l'envoi :** Vérifiez que votre archive est lisible et que votre code sera testable par le correcteur :  
lancez `verifier.py TPi_LOGIN1_LOGIN2.tar.gz` et vérifiez l'absence d'erreurs.
6. **Documentation :** A la racine de votre projet, 'make doc' doit générer une documentation valide (`doc/html/*.html`)
7. Noms des fichiers pour la phase d'implémentation :  
 $TP_1$  et  $TP_2$  `Dvector.h` et `Dvector.cpp`  
 $TP_3$  `Darray.h` et `Darray.cpp`, `Dvector.h` et `Dvector.cpp`, `Dmatrix.h` et `Dmatrix.cpp`  
 $TP_4$  `Point.h` et `Point.tpp`, `Triangle.h` et `Triangle.tpp`, `Maillage.h` et `Maillage.tpp`  
Aucun fichier complémentaire ne doit être requis (pas de `utilitaire.h/cpp` communs)
8. Nom des classes et des fonctions : Respecter scrupuleusement les énoncés.
9. Dans les phase d'implémentation, les questions marquées d'un \* sont utiles pour la vérification des suivantes. Accordez leur une attention toute particulière. Mal traitées, elles pourraient vous faire perdre des points sur les questions suivantes.
10. **Gestion des erreurs :** Si vous souhaitez gérer des erreurs (cas limites) non demandées dans l'énoncé, faites-le par l'intermédiaire d'exception (avec la méthode `throw`)
11. Phase d'analyse : Le fichier `analyse.txt` devra fournir les réponses à toutes les questions marquées d'un \* dans les énoncés.

1. doit contenir les cibles `all`, `check`, `doc`, `clean`