

Bilan du projet

Démarche

J'ai commencé par bien détailler le domaine, notamment les acteurs et les objets en question et les cas d'utilisation généraux auxquels je souhaitais répondre, et j'ai ensuite créé tous mes modèles ainsi que les tests unitaires associés. et enfin j'ai développé mes vues et mes templates conjointement aux tests fonctionnels, dans l'ordre dicté par les contraintes de modèle:

- d'abord traiter ce qui concerne le modèle utilisateur, qui est commun à toute l'application
- Après, le cas du référencement de l'équipement, qui est nécessaire pour traiter les deux dernières User stories
- ensuite l'enregistrement de feuille de score
- et enfin le calcul du carquois optimisé

conjointement, j'ai développé une CI qui vérifie que les tests unitaires et fonctionnels passent bien à chaque push peu importe la branche, et à chaque fusion dans la branche principale, ce qui m'a permis d'être sûr de ne pas importer de régression dans la branche principale.

J'ai déployé qu'à une étape du projet où il était très mature, quand tous les cas d'utilisation ont été implémentés.

Leçons tirées de l'expérience

sur ma pratique personnelle:

- Lorsqu'on commence par développer tous les modèles, il vaut peut-être mieux en créer des simplifiés, quitte à les compléter ensuite pour mieux coller aux cas d'utilisation. Ce serait une manière de faire plus agile, qui mette en avant la simplicité prônée dans les 12 pratiques de l'eXtreme programming.
- développer la partie CI plus tôt, en y intégrant un rapport de couverture de tests m'aurait permis de mieux suivre celle-ci
- développer une manière de déployer plus facilement et/ou plus automatiquement (soit dans la CI soit au travers d'un makefile) m'aurait rendu ce travail beaucoup plus facile
- j'ai profité de ce projet pour tester plusieurs technologies:
 - tailwindCSS
 - gitlabCI
 - poetry

Nous avons également beaucoup travaillé, avec mon mentor, à la manière de donner de la lisibilité sur mon travail, aboutissant à un certain nombre de pratiques, notamment:

- utiliser les outils de suivi de projet fournis par github
- référencement de l'Issue cernée dans chaque commit
- utiliser les mots-clés dans les commits pour fermer une issue, ou une pull requests si besoin
- réécriture d'historique avant de fusionner deux branches