

Fiche d'investigation de fonctionnalité

Fonctionnalité : Algorithme de recherche	Fonctionnalité #1
Problématique : Déterminer l'algorithme qui utilise le moins de ressource et qui s'exécute le mieux afin les recettes le plus rapidement au utilisateurs lors d'une recherche et également rendre le code le plus "green" possible.	

Option 1 : Boucle native (for) Dans cette option, nous utilisons des boucles natives pour effectuer la recherche d'un élément ajouté dans un input par l'utilisateur dans le tableaux des recettes. On affiche ensuite les recettes qui contiennent la recherche de l'utilisateur avec une boucle native qui affiche les cartes contenant les recettes.	
Avantages <ul style="list-style-type: none">⊕ Rapidité d'exécution de la recherche⊕ Code plus logique à mettre en place (Habitue d'utilisation)	Inconvénients <ul style="list-style-type: none">⊖ Plus de lignes de code⊖

Option 2 : Boucle d'ordre supérieur (filter, map...) Dans cette option, nous utilisons des boucles d'ordre supérieur pour effectuer la recherche d'un élément ajouté dans un input par l'utilisateur dans le tableaux des recettes. On affiche ensuite les recettes qui contiennent la recherche de l'utilisateur avec une boucle d'ordre supérieur qui affiche les cartes contenant les recettes.	
Avantages <ul style="list-style-type: none">⊕ Moins de lignes de code⊕ Code plus lisible⊕ Plus rapide à mettre en place	Inconvénients <ul style="list-style-type: none">⊖ Moins rapide à l'exécution

Solution retenue : Nous retenons l'approche Boucle Native car elle a une meilleure rapidité d'exécution et c'est la problématique de cet algorithme. On veut le code qui s'exécute le plus rapidement possible. On pourrait également prendre en compte le nombre de ligne de code mais finalement il y'a pas plus de fichiers en fonction du type de boucle choisi.

Annexes



