

Fiche d'investigation de fonctionnalité

- **Fonctionnalité** : Algorithme de recherche
- **Problématique** : Afin de se démarquer de la concurrence, nous cherchons à développer un moteur de recherche fluide avec une recherche se basant sur 3 critères (Ingrédients, Appareils et Ustensiles).
- **Option 1** : Algorithme de recherche se basant sur les boucles natives (while, for...).
 - **Description** : Une boucle for répète des instructions jusqu'à ce qu'une condition donnée ne soit plus vérifiée:

```
for ([expressionInitiale]; [condition]; [expressionIncrément]) {  
    instruction  
}
```
 - **Avantages** :
 - Rapidité et performance
 - Possibilité d'ajouter une fonction callback si besoin.
 - **Inconvénients** :
 - L'algorithme utilisant les boucles natives est plus long à écrire et moins lisible.
 - Lignes de code plus denses
 - Présence d'index et d'incrément.
- **Option 2** : Algorithme de recherche se basant sur les méthodes de l'objet array (foreach, filter, map, reduce).
 - **Description** : La méthode ForEach() permet d'exécuter une fonction callback sur chaque élément d'un tableau dans l'ordre croissant de l'indice. ForEach ne modifie pas le tableau sur lequel elle est appelée.
 - **Avantages** :
 - Pratique car elle est disponible par défaut.
 - Concise car elle a moins de ligne de code.
 - **Inconvénients** :
 - L'appel d'une fonction callback sur chaque élément ralentit la performance sur un tableau de taille conséquente.
- **Solution retenue** : L'option 1 car elle est plus performante d'après le benchmark Javascript. Tout en gardant la logique même du filtrage malgré la lisibilité et la longueur du code.