

## Fiche d'investigation de fonctionnalité

Fonctionnalité: Search	
Problématique: Dans le but d'une recherche fluide et dans un soucis d'optimisation, vous voulez un algorithme qui soit le plus performant possible.	

Option 1: Algorithme de recherche utilisant la programmation fonctionnelle Dans ce cas, l'algorithme de recherche va se baser sur les éléments de la méthode array(filter, reduce, map etc...) Le principal avantage à ces méthodes sont leurs maintenabilité car facile à lire. De l'autre côté l'inconvénient principal reste leur performance car plus d'étapes sont faites pour renvoyer une réponse.	
Avantages : <ul style="list-style-type: none"><li>• Facile à mettre en œuvre</li><li>• Maintenance</li></ul>	Inconvénients: <ul style="list-style-type: none"><li>• Performances</li></ul>
Performances lors du jsBench : 3163.9 ops	

Option 2 : Algorithme de recherche utilisant les boucles natives Dans ce cas, l'algorithme de recherche va utiliser des boucles natives(for, while). L'avantage principal sera les performances. En revanche, moins simple à mettre en œuvre car plus de paramètres à entrer.	
Avantages: <ul style="list-style-type: none"><li>• Performances</li><li>• Meilleurs contrôles des itérations</li></ul>	Inconvénients: <ul style="list-style-type: none"><li>• Mise en œuvre</li><li>• Maintenance (lisibilité)</li></ul>
Performances lors du jsBench : 1413.13 ops	

Solution retenue : Étant donné que l'objectif principal de l'algorithme est la performance, la solution retenue sera "Algorithme de recherche utilisant la programmation fonctionnelle" car il obtient de meilleures performances lors du jsBench.
---

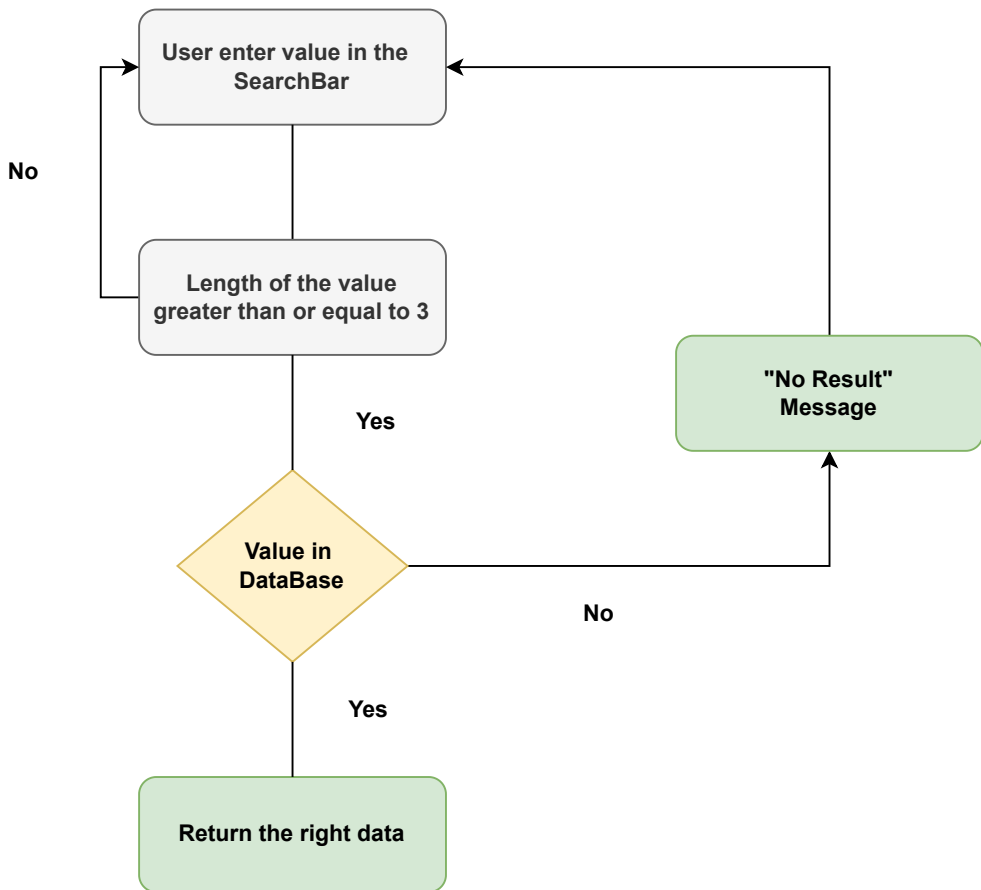


Figure 1 - Diagramme des deux algorithmes