

## Les Petits Plats

### Fiches d'investigation de fonctionnalité :

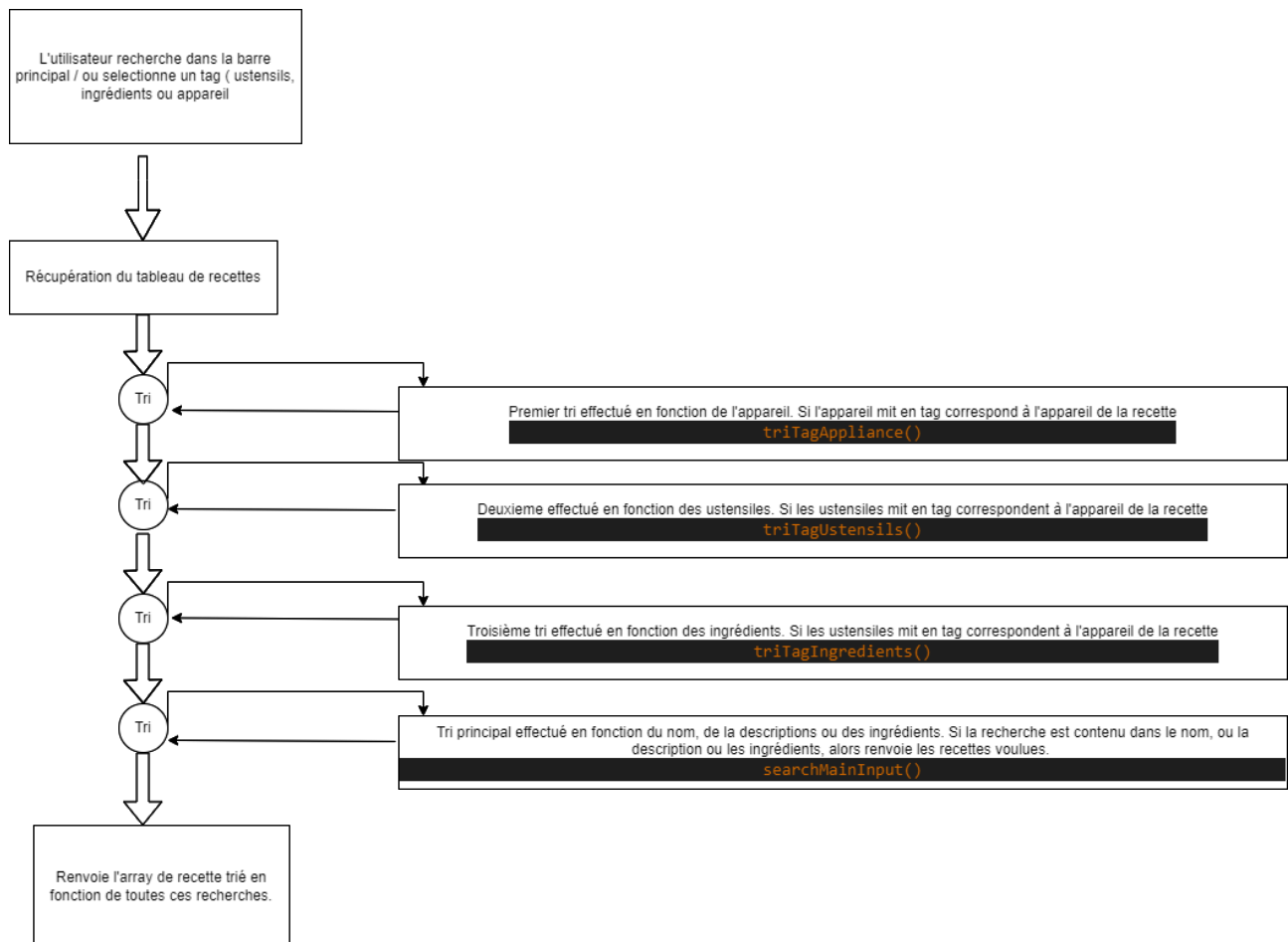
Fonctionnalité : Algorithme de recherche	Fonctionnalité #1
<p>Problématique :</p> <p>Permet de trier les recettes en prenant plusieurs paramètres en comptes :</p> <ul style="list-style-type: none"><li>1) Un champ de recherche principal</li><li>2) 3 Champ de selection ( ustensiles, ingrédients, appareil)</li></ul> <p>Afin de chercher à fournir le meilleur service à l'utilisateur, nous avons souhaité tester deux implémentation de l'algorithme de recherche de recettes au sein de l'application web Les Petits Plats</p> <p>Si aucun tri n'est effectué, la liste de toutes les recettes s'affiche.</p>	

<b>Option 1 :</b> Avec des boucles natives for /et les if	
Avantages : simple à mettre en place	Inconvénients : <ul style="list-style-type: none"><li>1) Lenteur d'execution</li><li>2) Fonctions longues, difficilement maintenables.</li></ul>

<b>Option 2 :</b> Algorithme avec des méthodes sur les array / JavaScript ES6	
Avantages : <ul style="list-style-type: none"><li>1) Performant</li><li>2) Facilement maintenable</li><li>3) Fonctions faciles à comprendre</li></ul>	Inconvénients : <ul style="list-style-type: none"><li>1) Possibilité de difficulté dans la compréhension à première vue des fonctions ES6</li></ul>

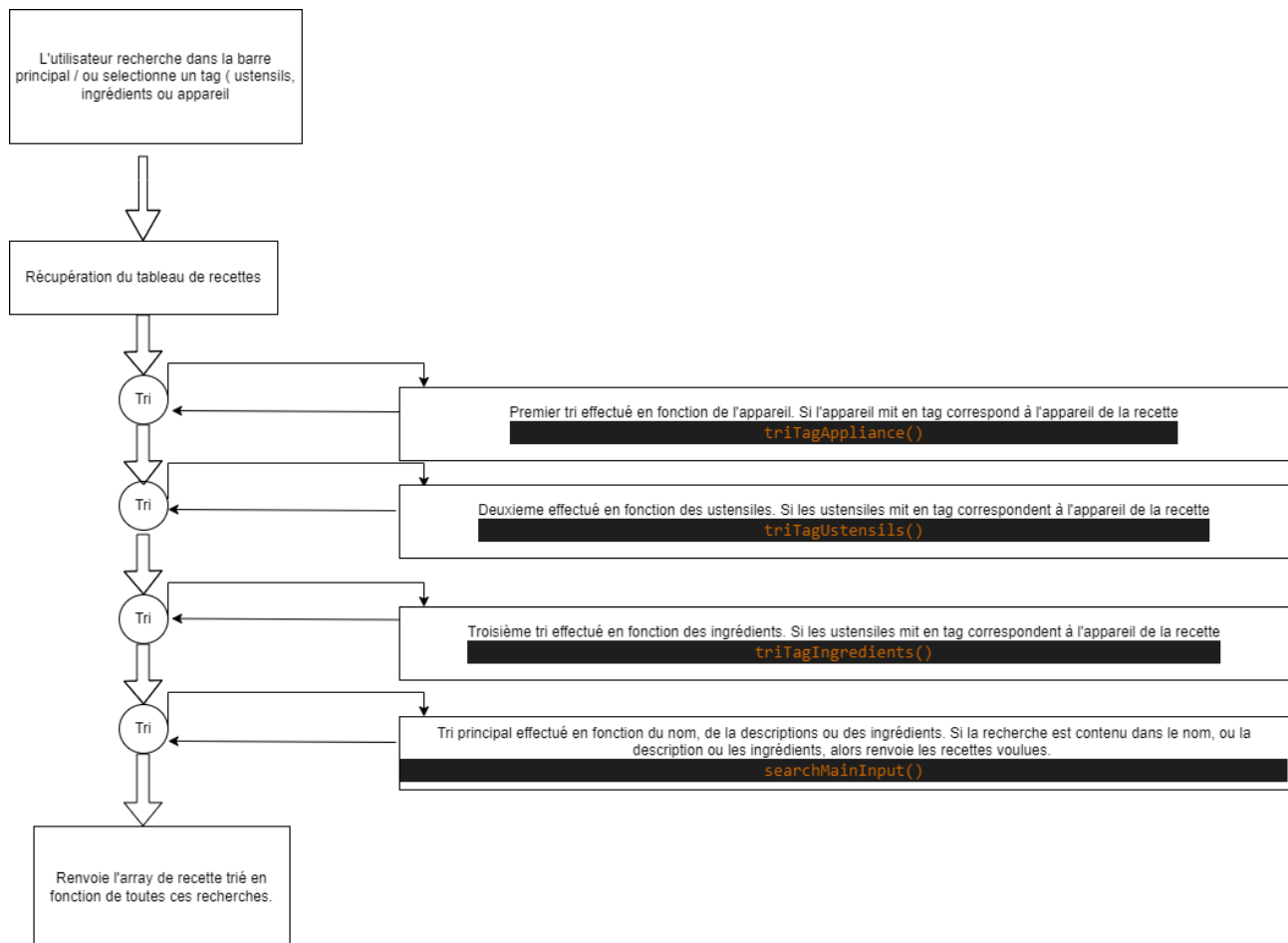
## Algorithme 1 :

### Fonctions utilisant les boucles natives

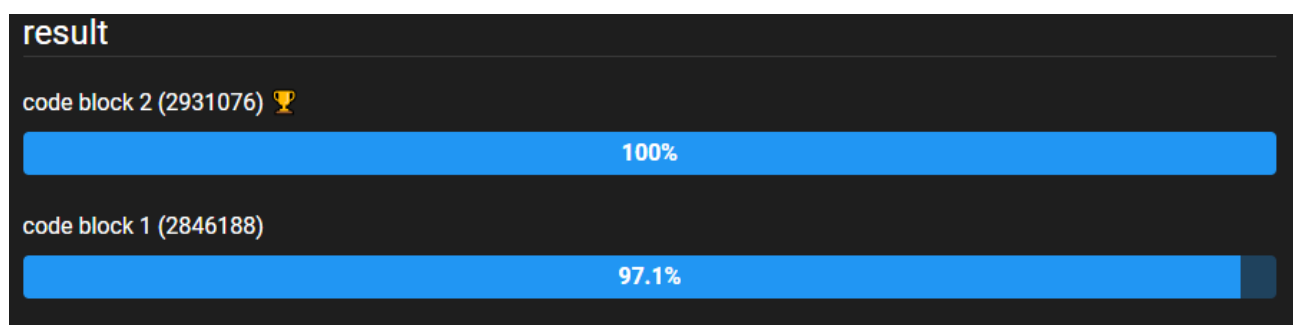


## Algorithme 2 :

### Fonctions utilisant les méthodes sur les listes ( array)



Réultats des comparaisons sur <https://jsben.ch/OP6Bb>



**Solution retenue :** Option 2 car l'algorithme est plus performant et plus facilement maintenable que le premier.