

Fiche d'investigation de fonctionnalité

Fonctionnalité : Algorithme de Recherche	Fonctionnalité #1
Problématique : Créer un algorithme de recherche qui soit à la fois rapide et performant pour le confort de l'utilisateur.	

Option 1 : Recherche en programmation fonctionnelle Dans cette option, on utilise les boucles foreach pour chaque mot clé renseigné dans les champs ingrédients et ustensiles.	
Avantages ⊕ Lisibilité du code	Inconvénients ⊖ Utilise plus de ressources
Cas n°1 : L'utilisateur remplit uniquement le champ de recherche Cas n°2 : L'utilisateur laisse le champ de recherche vide et sélectionne un tag Cas n°3 : L'utilisateur sélectionne un tag et remplit le champ de recherche	

Option 2 : Recherche en programmation native Dans cette option, on utilise les boucles for pour chaque mot clé renseigné dans les champs ingrédients et ustensiles.	
Avantages ⊕ Rapidité d'exécution	Inconvénients ⊖ Code moins lisible et donc moins maintenable
Cas n°1 : L'utilisateur remplit uniquement le champ de recherche Cas n°2 : L'utilisateur laisse le champ de recherche vide et sélectionne un tag Cas n°3 : L'utilisateur sélectionne un tag et remplit le champ de recherche	

Solution retenue : Nous avons donc retenu l'approche 2 de la "Recherche en programmation native" avec usage des boucles for, l'objectif premier étant la rapidité d'exécution de l'algorithme de recherche.

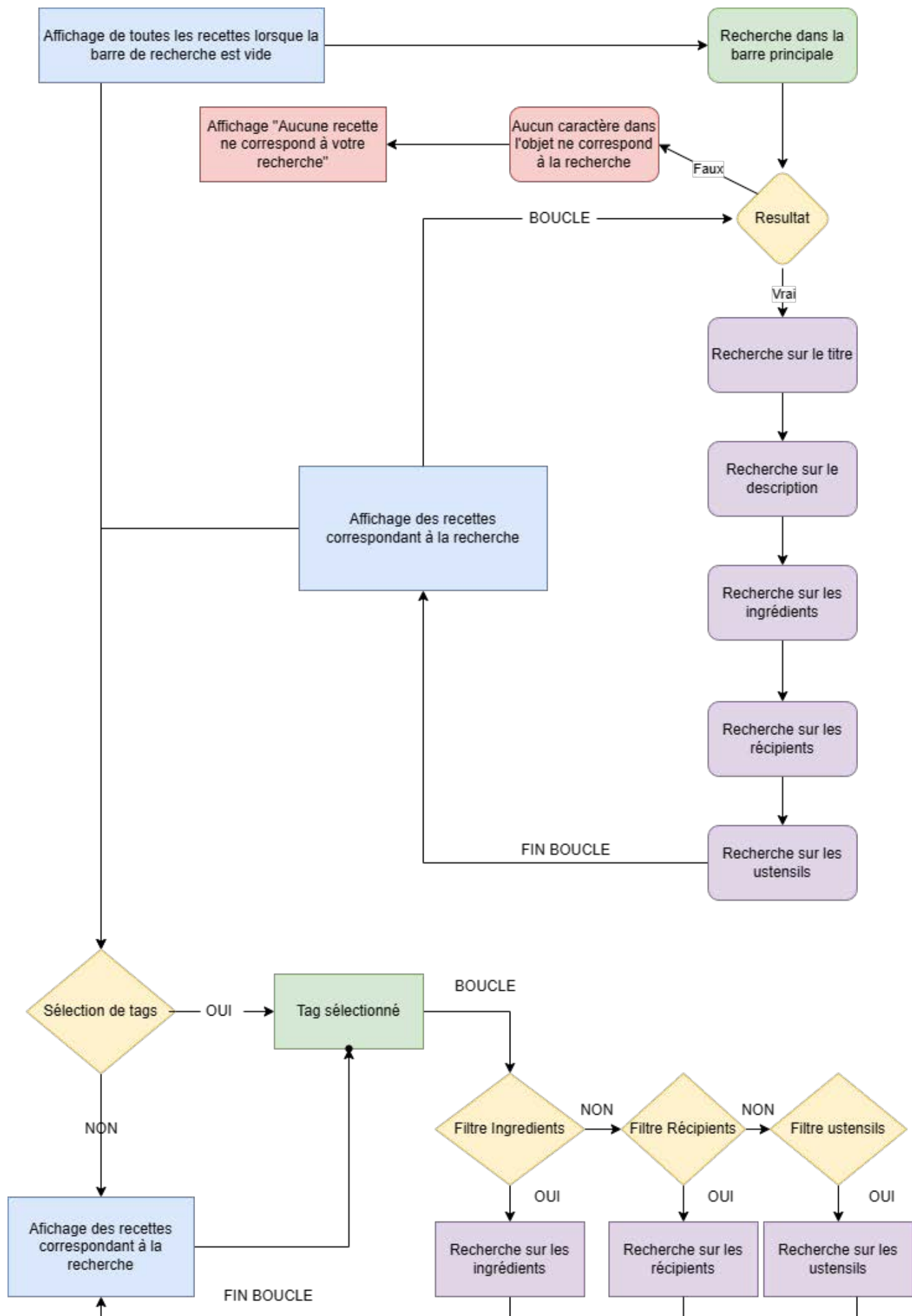


Figure 1 - Recherche globale

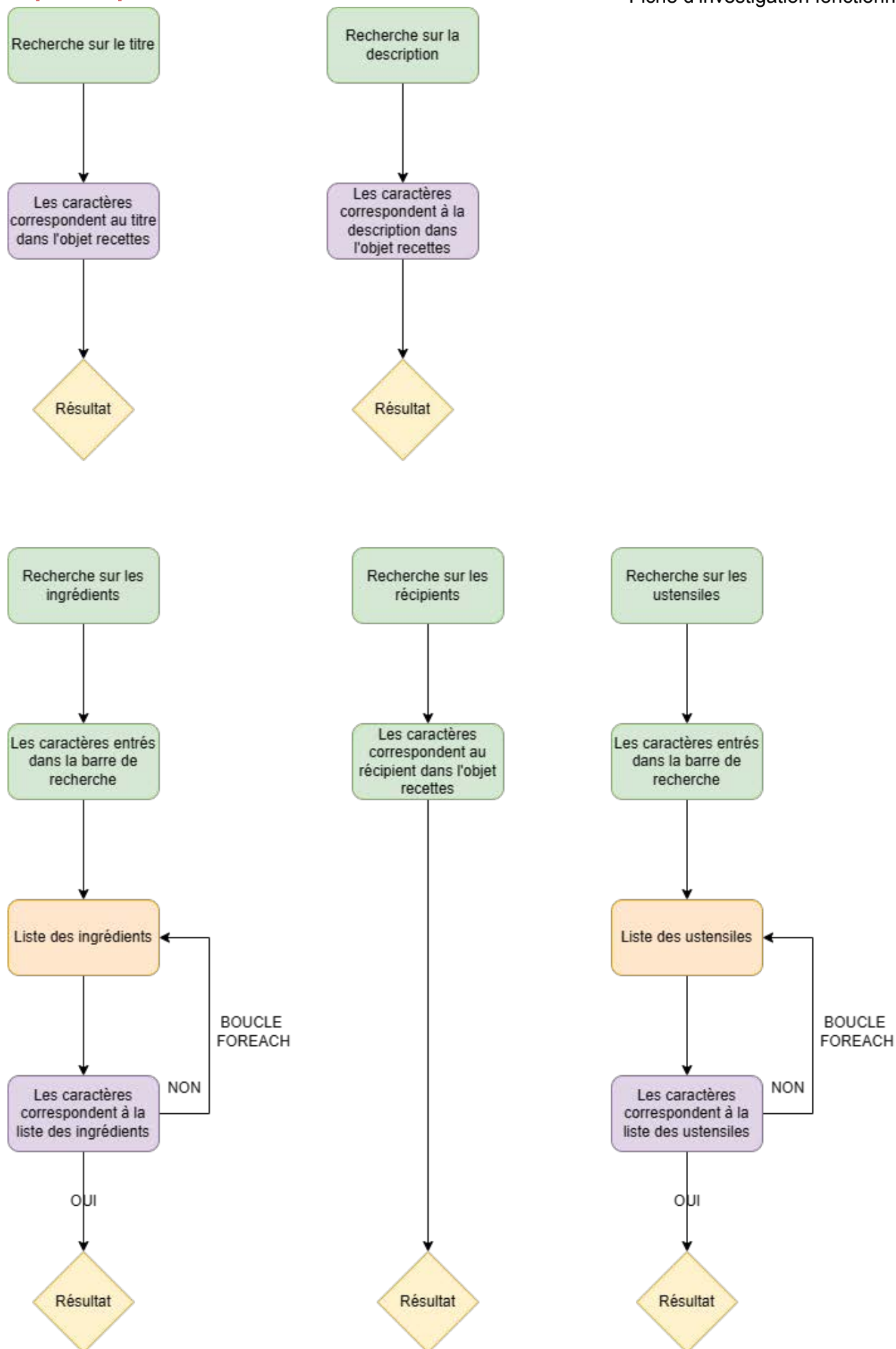


Figure 2 : Approche "Recherche en programmation fonctionnelle"

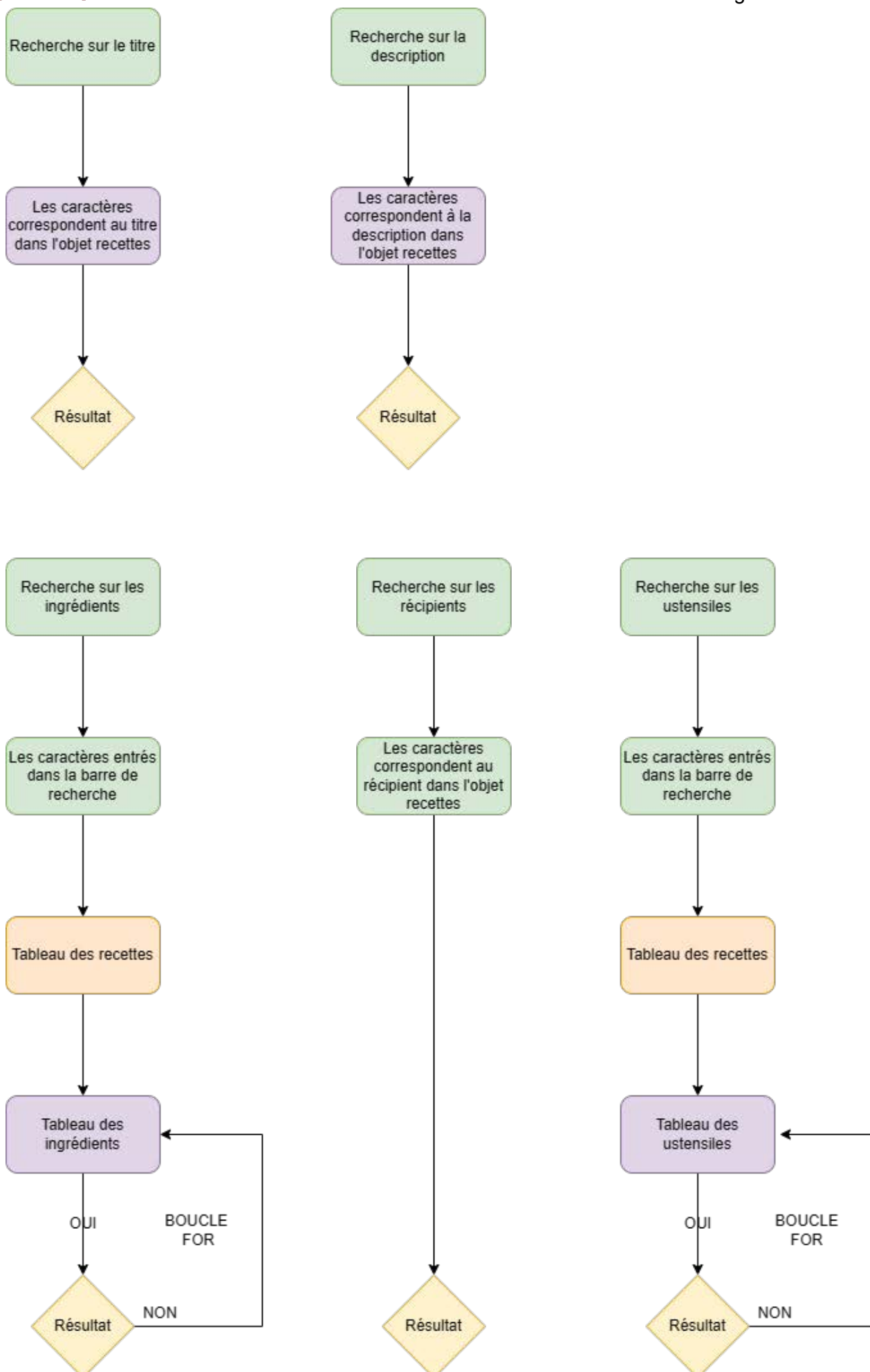
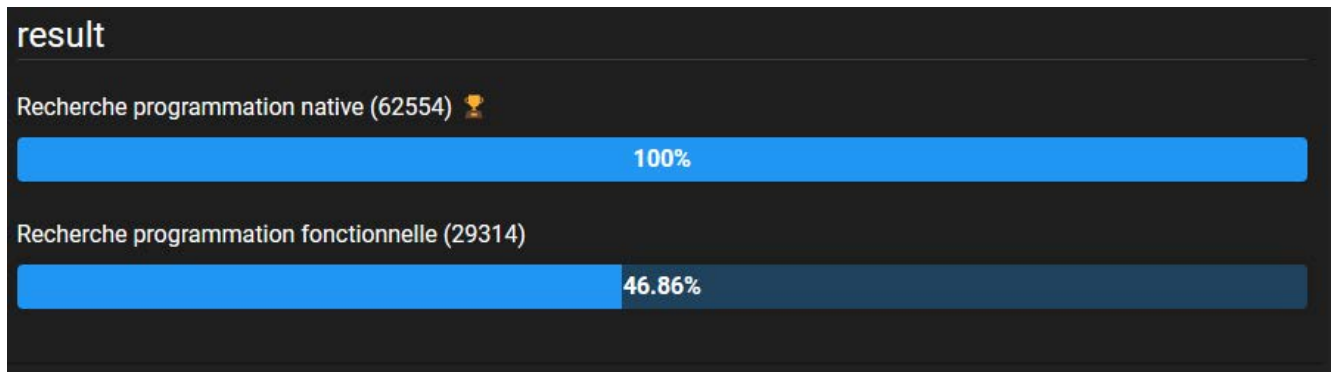


Figure 3 : Approche "Recherche en programmation native"

Résultat des test

Cas recherche du mot "coco"



1	Recherche en programmation native	4,488 ops/s
2	Recherche en programmation fonctionnelle	2,148 ops/s