

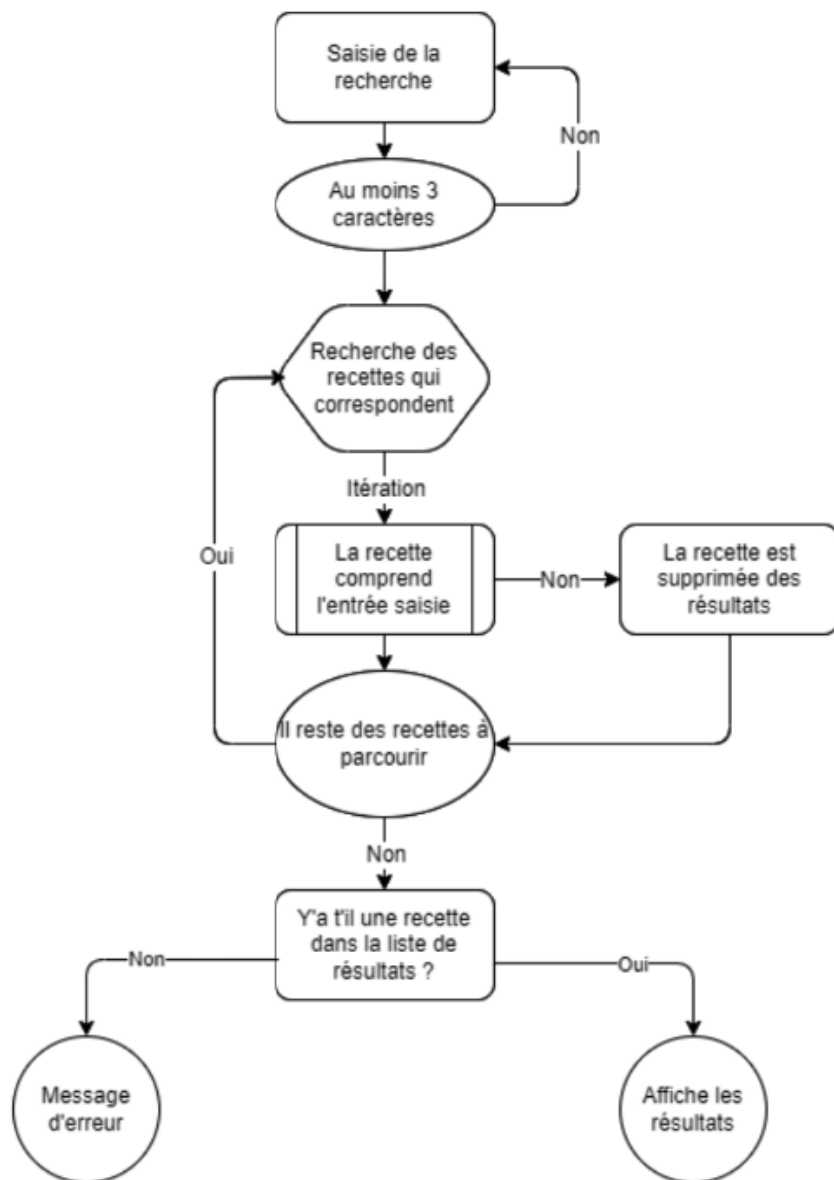
Fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche principale	Fonctionnalité #1
Problématique : Afin de répondre aux besoins des utilisateurs, la recherche doit être la plus rapide possible.	

Option 1 : Programmation fonctionnelle Cette option utilise les méthodes de l'objet array (foreach, filter, map, reduce)	
Avantages : <ul style="list-style-type: none">• Code plus compréhensible• Peu de changements en cas de nouveaux filtres	Inconvénients : <ul style="list-style-type: none">• Incompatibilité avec certains navigateurs

Option 2 : Boucles natives Cette option utilise les boucles while et for	
Avantages : <ul style="list-style-type: none">• Compatibilité avec tous les navigateurs	Inconvénients : <ul style="list-style-type: none">• Code plus complexe, moins logique• Beaucoup de changements en cas de nouveaux filtres

Solution retenue : Nous avons donc retenu l'approche fonctionnelle. Suite aux tests réalisés (<i>Annexe</i>), nous avons pu constater que la programmation fonctionnelle est plus rapide que les boucles natives. De plus, la mise en place de nouveaux filtres étant simplifiée, maintenir le site sera facilité en cas de mises à jour.



enter test suite name

v1

- by

-

enter test suite description

+ Setup HTML - click to add setup HTML

+ Setup JS - click to add setup JavaScript

method array

finished

1814668.53 ops/s \pm 3.09%

Fastest

```
card.innerHTML = templateRecipes; // ajout du template de la carte de recette
card.querySelector("h1").innerHTML = this.name; // ajout du nom de la recette
card.querySelector(".time").innerHTML = this.time + " min"; // ajout du temps de préparation
card.querySelector(".description").innerHTML = this.description; // ajout de la description
this.ingredients.forEach(ingredient) => {
  // pour chaque ingrédient
  const li = document.createElement("li"); // création de la liste
  li.innerHTML = templateIngredient; // ajout du template de l'ingrédient
  li.querySelector(".ingredient").innerHTML = ingredient.ingredient; // ajout de l'ingrédient
  li.querySelector(".quantity").innerHTML =
    ingredient.quantity !== undefined ? `${ingredient.quantity}` : ""; // ajout de la quantité si elle
```

☐ DEFER

boucle native

finished

1745222.97 ops/s \pm 3.19%

3.83 % slower

```
getCard() {
  // création de la carte de recette
  const card = document.createElement("article"); // création de l'article
  card.classList.add("recipe-card"); // ajout de la classe recipe-card
  card.innerHTML = templateRecipes; // ajout du template de la carte de recette
  card.querySelector("h1").innerHTML = this.name; // ajout du nom de la recette
  card.querySelector(".time").innerHTML = this.time + " min"; // ajout du temps de préparation
  card.querySelector(".description").innerHTML = this.description; // ajout de la description
  for (let i = 0; i < this.ingredients.length; i++) {
    // pour chaque ingrédient
    const li = document.createElement("li"); // création d'une liste
```

☐ DEFER

+ Test Case - click to add another test case

+ Teardown JS - click to add teardown JavaScript

+ Output (DOM) - click to monitor output (DOM) while test is running

▶ RUN again