

Fiche d'investigation de fonctionnalité

Fonctionnalité : Rechercher des recettes

Problématique :

Rechercher et filtrer des recettes selon l'entrée utilisateur.

Option 1:

Utilisation des fonctions de prototype array

Emploi de la méthode **foreach**,

permettant de faire des itérations dans les recettes.

L'algorithme vérifie la correspondances entre les données, et l'entrée saisie par l'utilisateur via un champ de recherche.

Il se base sur le nom de la recette, les ingrédients de la recette, ou sa description.

Si la recherche correspond, seules les recettes correspondantes seront affichées.

Avantages:

- Maintenable
- Compréhensible

Inconvénients:

- Exécution du code parfois plus lent

Conditions d'exécution: Entrée utilisateur supérieure ou égale à 3 caractères

Option 2:

Utilisation de la programmation native

Emploi de la boucle **"for"**,

permettant une fois de plus de faire des itérations dans les recettes.

L'algorithme vérifie la correspondances entre les données, et l'entrée saisie par l'utilisateur via un champ de recherche.

Les recettes seront alors mis à jour pour n'afficher que les recettes correspondantes à la saisie utilisateur.

Avantages:

- Exécution du code parfois plus rapide

Inconvénients:

- Difficilement Maintenable
- Moins compréhensible

Conditions d'exécution: Entrée utilisateur supérieure ou égale à 3 caractères

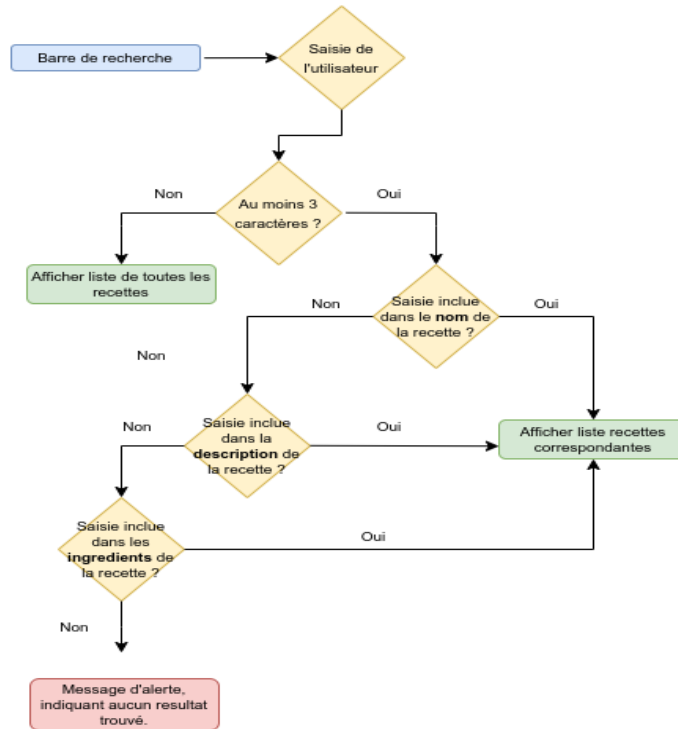
Solution retenue:

L'option 2 utilisant la programmation native, paraît en effet plus rapide, cependant ce gain de rapidité n'est pas assez conséquent par rapport à l'option 1 utilisant les fonctions de prototype array.

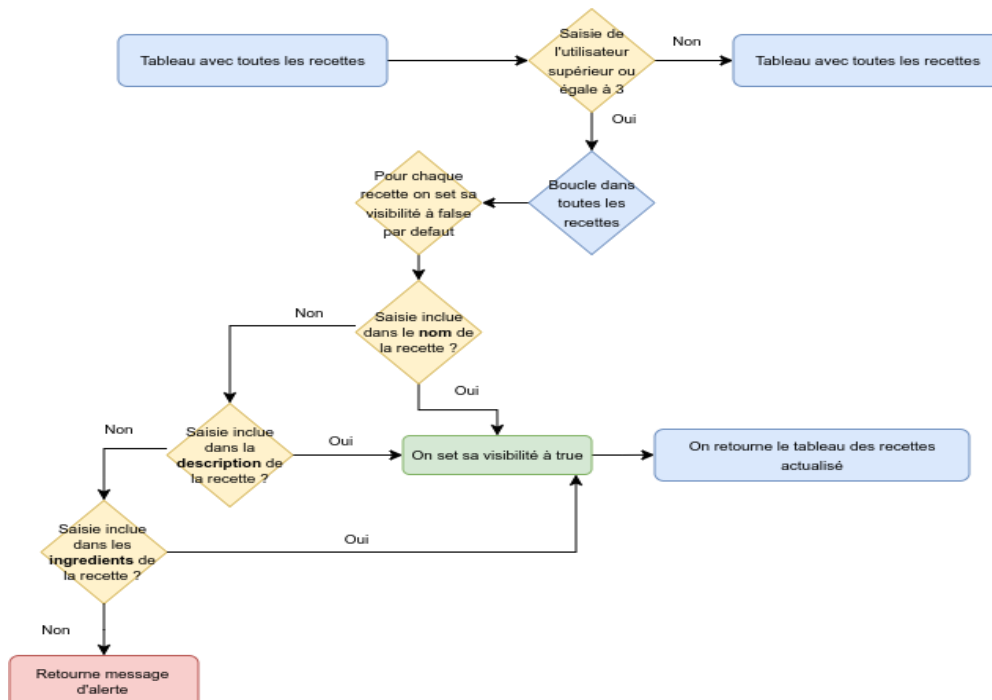
Le choix se porte donc sur l'option 1, utilisant les fonctions de prototype array; **foreach**.

Annexe 1

Schema fonctionnel



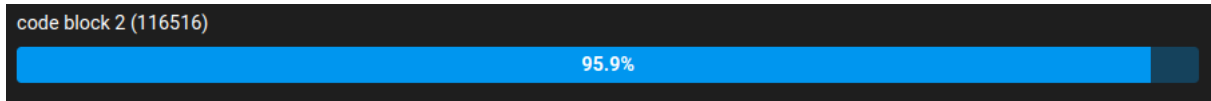
Schema technique



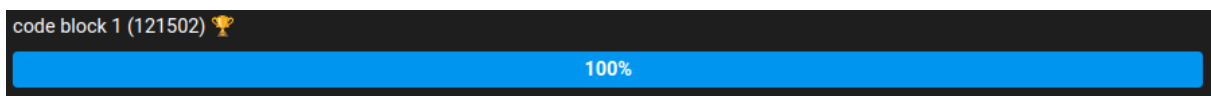
Annexe 2: Test JsBen.ch

1000 Recettes: Foreach Win

For

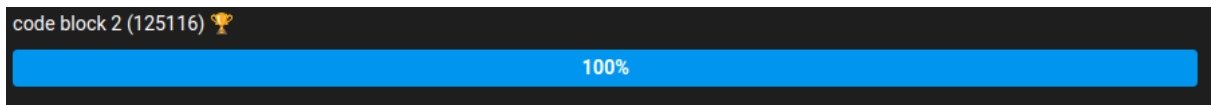


Foreach

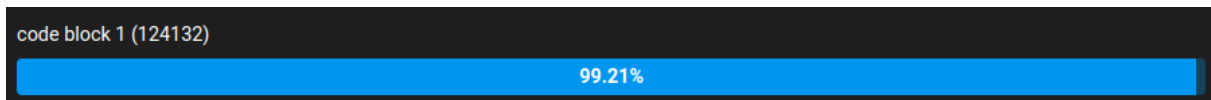


500 Recettes: For Win

For

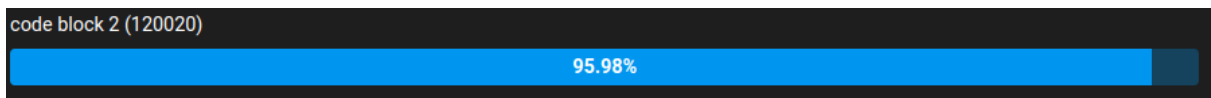


Foreach

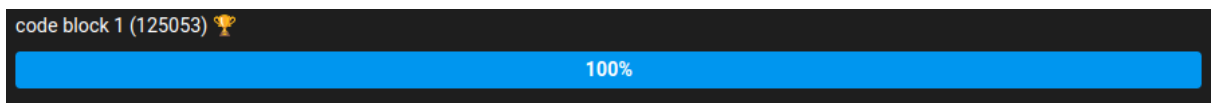


300 Recettes: Foreach Win

For



Foreach



50 Recettes For Win

For

code block 2 (122186) 🏆

100%

Foreach

code block 1 (119738)

98%

Bilan des tests

Nb recettes	For	Foreach	Résultat
50	100%	98%	For: 2% Faster
300	95.98%	100%	Foreach: 4.02% Faster
500	100%	99.21%	For: 0.79% Faster
1000	95.9%	100%	Foreach: 4.1% Faster

Conclusion

Les résultats sont assez équilibrés, cependant quand le résultat à un plus gros écart, **Foreach** est le meilleur résultat.

En prenant en compte celà, et que foreach soit bien plus lisible, compréhensible et maintenable, le choix final est donc sur l'utilisation de **Foreach**.