

# Projet de substitutions d'aliments pour le restaurant « **Ratatouille** » de la startup « **Pur Beurre** » .

L'objectif étant de créer un programme qui va interagir avec la base de données alimentaire internationale « **Open Food Facts** ».

Les stratégies de développement que j'ai choisies sont issues de méthodes bien connues.

La première méthode est le « **Readme Driven Development** » qui consiste à créer un fichier texte au format Markdown pour lister les principales fonctionnalités du programme.

La méthode suivante vient d'une méthodologie agile, « le Tableau Trello » .

Les différentes tâches et sous-tâches à accomplir sont divisées en user stories regroupées dans un tableau dont voici le lien : [Tableau TRELLO](#)

Le choix du langage de code s'est orienté vers Python 3 pour sa philosophie simple de codage, ses nombreuses bibliothèques facilitant l'accès aux données, sa documentation bien fournie et sa communauté de soutien importante.

Quant au choix de la base de données, cela s'est porté sur le langage SQL et son Système de Gestion de Base de Données : MySQL lui aussi pour sa large communauté et sa documentation assez détaillée.

Pour le fichier « **Readme** » pas de problématique particulière

- Mise en place et utilisation de la structure du programme
- Technique de construction du projet
- Installation du programme pour le projet

Concernant les différentes tâches du tableau « **TRELLO** »

- Modèle physique de la base de données  
seule la table d'enregistrement pour les produits choisis a posé soucis,  
la solution adoptée : création d'une table de liaison entre la table produit et la table produit (ip produit choisi, id produit substitué (choisi))
- Trouver les liens de catégorie produit dans la base de données Open Food Facts  
long et fastidieux travail de recherche sur la base Open Food Facts

## Différentes tâche du tableau « **TRELLO** » suite...

- Création base de données
  - Exportation et conversion du modèle physique de la base de données (MPD) en script SQL enregistré dans fichier au format sql
- Code du script écrit en python 3 en plusieurs étapes
  - Création des différents menus chiffrés de choix (1<sup>er</sup> module)
    - menu de choix de catégorie produit
    - menu de choix de produit selon la catégorie choisi
    - menu de produit substitué de meilleures qualité nutritionnelle que le produit choisi
  - Création de la gestion de la base de données (2eme module)
    - remplissage des différentes tables de la base de données
    - effacement des tables de la base de données

Un des soucis majeur dans l'écriture du code à été de gérer l'accès multiple à la base de données. Ce problème a été résolu en créant une classe contenant les arguments de connexion et de fait chaque connexion s'exécute avec une instance de cette classe.

L'autre soucis d'écriture a été de garder un code répondant le plus possible aux conventions PEP tout en restant fonctionnel. Pour répondre à ces contraintes un troisième module à vu le jour.

Pour Conclure, le code se compose d'un module principal « **main.py** » qui contient la classe pour la connexion à la base de données ainsi que le démarrage du script

1) teste si les tables sont vident et le remplit le cas échéant avec des données JSON locale (pas besoin d'internet pour la mise en route du script)

2) Affiche le menu général de choix catégorie produits...

Le module1 « **prod\_processing.py** » qui gère tout la gestion des menus de choix...

Le module2 « **db\_processing.py** » gère le traitement sur la bas de données...

Le module3 « **prod\_methode.py** » est là surtout pour alléger la structure de code du module 1 afin de garder une cohérence selon la convention PEP...

Le projet est hébergé sur github : [Lien Hébergement Github](#)