

Développement d'Applications Python – Projet 05

« Utilisez les données publiques de l'OpenFoodFacts »

1. Lien vers le code source du projet sur GitHub

https://github.com/etienne86/oc_p05_open_food_facts

2. Reformulation succincte du sujet

L'objectif de ce projet est de concevoir une application qui propose, à partir d'un aliment choisi par l'utilisateur, un aliment de substitution réputé meilleur du point de vue nutritionnel.

Le programme s'appuie sur une base de données locale MySQL, dont les produits sont issus de l'API d'OpenFoodFacts.

Le programme fonctionne dans le terminal de l'utilisateur.

L'utilisateur a la possibilité d'enregistrer le substitut proposé par le programme.

3. Gestion du projet

3.1. Regard critique sur la base de données d'OpenFoodFacts

La base de données mise à disposition par OpenFoodFacts contient un nombre très grand de lignes (une ligne représentant un produit), croissant chaque jour.

De plus, étant renseignée par des volontaires bénévoles et n'étant pas vérifiée en totalité, la qualité des informations saisies n'est pas complètement fiable.

Concernant certains produits, les informations utiles au programme sont manquantes : j'ai donc décidé de ne pas utiliser ces produits dans l'application, afin de ne conserver que les aliments dont les renseignements nécessaires sont connus.

3.2. Interface avec l'API d'OpenFoodFacts

L'import des données des produits depuis l'API d'OpenFoodFacts est nécessaire au moment de l'installation, plus précisément lors du remplissage de la base de données.

J'ai effectué des tests de transfert de données en utilisant des requêtes HTTP, avec une connexion entrante ayant une vitesse de 750ko/s en moyenne, pour importer respectivement 100, 250, 500 et 1000 produits. D'après mes tests, le temps de transfert est proportionnel au nombre de produits importés, ce qui implique un temps projeté de transfert d'environ 5h et 45min pour 800 000 produits. J'ai donc choisi de ne pas tout importer, mais de limiter mon application à **9 catégories d'aliments**, avec **au maximum 1000 produits par catégorie**.

3.3. Lancement du projet

Contrairement au projet 03 où je n'avais pas utilisé de méthodologie de gestion de projet particulière, sur ce projet j'ai travaillé en mode agile et j'ai suivi l'orientation proposée en utilisant également la méthodologie du *Test Driven Development*.

En effet, j'ai commencé par ébaucher le document README.md dans lequel je présente la future application en expliquant notamment comment elle est construite (pour les développeurs potentiellement amenés à lire le code source) et comment elle

s'utilise (pour l'utilisateur du programme). Au départ, ce document contenait surtout les titres de rubriques, et quelques généralités sur l'application.

En parallèle, j'ai commencé à créer des cartes Kanban dans un tableau agile de gestion de projet, avec des *user stories* et des *stories* techniques. Le tableau se trouve à cette adresse : <https://trello.com/b/Q6r47F1d/e-barbier-oc-da-py-open-food-facts>.

3.4. Développement du projet

J'ai élaboré une première version du diagramme des classes (pour le programme avec Python), puis du modèle physique de données (pour la base de données avec SQL). De nombreuses itérations de ces deux documents sont venues les enrichir, au fur et à mesure de l'avancée du projet, notamment lors de l'intégration du code Python, module par module.

De même, j'ai procédé par itérations successives pour compléter alternativement le document README.md et mettre à jour le tableau agile en ajoutant de nouvelles cartes, car de nouvelles fonctionnalités s'avéraient utiles (pour les *user stories*) ou de nouvelles activités étaient nécessaires pour progresser dans le développement du projet (pour les *stories* techniques).

4. Retour d'expérience

Lors du développement de mon projet, j'ai relevé plusieurs éléments clés :

- La découverte de la base de données d'OpenFoodFacts et de son API a été fastidieuse : j'ai notamment investi beaucoup de temps à essayer de comprendre le mode de fonctionnement de l'API, malgré la documentation fournie
- Le choix des modules externes :
 - o Concernant l'interaction avec la base de données MySQL, mon choix s'est porté sur **mysql.connector** en raison de la facilité d'utilisation du module
 - o Concernant l'interaction avec l'API d'OpenFoodFacts, le module **requests** m'a également paru rapide et simple à prendre en main
 - o Concernant la façon de présenter l'application dans le terminal, sans interface graphique, afin que le programme reste agréable à utiliser malgré l'aspect basique de la console, j'ai préféré présenter les données sous forme de tableau, avec le module **texttable**. En effet, cela m'a semblé plus lisible pour l'utilisateur que ma solution initiale qui n'affichait que du texte à l'écran.
- Le choix de l'algorithme de sélection du substitut : j'ai considéré qu'un substitut se trouve dans la même catégorie, celle de la recherche de l'utilisateur. Ainsi, tous les autres aliments de la catégorie sont potentiellement des substituts : il n'y a pas de distinction entre eux. Le substitut proposé est alors celui qui a le nutri-score numérique français (la clé **nutrition-score-fr_100g** dans la base de données) le plus faible. Puisque qu'il y a peu de différences avec *nutrition-score-uk_100g*, j'ai décidé d'ignorer cette dernière valeur, alors que je comptais initialement l'utiliser pour départager d'éventuels ex aequo : ainsi, dans une catégorie, les ex aequo potentiels sont départagés de façon aléatoire.
- La gestion de **projet en mode agile** a été ressentie comme un outil aidant, plutôt qu'une contrainte, et m'a permis de mieux organiser mon travail en le planifiant et en améliorant ma visibilité sur les activités à venir.
- De même que pour le point précédent, j'ai apprécié utiliser **Git et GitHub** et je m'en suis servi davantage que lors du projet 03. J'ai notamment créé une seconde branche pour tester de nouvelles fonctionnalités, avant de la fusionner avec la première et poursuivre le projet.