

FoodTrack

BAGUINI Josselin - BOUKEZZOULA Sami - MARTIN Eliot - MOYAUX Adrien

Introduction

Nous avons décidé de créer l'application mobile **FoodTrack**, dont l'objectif est d'afficher l'impact environnemental des produits alimentaires achetés par les consommateurs au travers d'un indicateur (l'[éco-score](#)) au fonctionnement analogue à celui du [nutri-score](#). L'idée est de développer un outil simple d'utilisation pour que les consommateurs puissent obtenir des informations chiffrées et fiables (par exemple l'empreinte carbone ou la provenance des ingrédients) des produits en scannant simplement leur code-barres.



Nous nous sommes inspirés de l'initiative déjà existante développée par l'ONG [Open Food Facts](#), qui a pour but de rendre plus transparente la composition des produits alimentaires et leur provenance. Proposant déjà un outil de calcul de l'éco-score, cette association ne fournit cependant pas systématiquement les détails de ce dernier et n'est pas non plus experte dans ce domaine, ce pourquoi FoodTrack pourra être un bon complément à leur initiative.

Notre démarche s'est articulée autour de 2 axes principaux :

- Synthétiser l'empreinte environnementale du produit à l'aide d'un éco-score facilement compréhensible par le consommateur
- Rendre la chaîne logistique plus transparente pour le consommateur, en traçant le parcours des denrées alimentaires du producteur au consommateur

Extraction des données

● Description des bases de données

Pour obtenir les informations générales d'un produit (liste des ingrédients, origine, masse...), nous avons utilisé l'[API Open Food Facts](#). Interrogeable à partir de la seule connaissance du code-barres du produit, elle renvoie un fichier JSON contenant entre autres ces informations. L'avantage d'utiliser cette API consiste à toujours bénéficier des mises à jour les plus récentes de la base de données Open Food Facts (constituée de millions de produits et mise à jour quotidiennement), sans avoir à les gérer : c'est donc une extraction incrémentielle, au fur et à mesure des requêtes des utilisateurs. Un inconvénient que nous avons relevé est toutefois la dépendance de notre application au fonctionnement de cette API (notamment en cas de maintenance). Ceci pourrait être résolu en effectuant un export de cette base de

données (de plus de 30 Go) dans MongoDB et en la mettant à jour quotidiennement à l'aide d' "exports delta".

Les données relatives à l'impact environnemental des ingrédients utilisés dans les produits alimentaires consommés en France sont issues de la base de données publique [AGRIBALYSE](#), développée entre autres par l'ADEME. Chaque ingrédient est repéré par un code [CIQUAL](#) (attribué par l'ANSES). Celui-ci permet de faire le lien entre la base de données AGRIBALYSE et la liste des ingrédients renvoyée par l'API Open Food Facts (le code CIQUAL est donné pour chaque ingrédient). Nous avons téléchargé la base de données AGRIBALYSE sous la forme d'un fichier Excel (d'environ 2 Mo) que nous avons d'abord converti en fichier CSV afin de l'intégrer dans une base de données PostgreSQL. En effet, le format standardisé des données issues de la base de données AGRIBALYSE et sa faible fréquence de mise à jour (tous les 2 ans) permet d'exploiter le caractère relationnel des bases de données PostgreSQL.

Enfin, afin d'avoir plus d'informations sur le transport, nous avons supposé que nous pourrions être en partenariat avec un transporteur (comme CMA-CGM). Ce dernier nous donnerait ainsi accès à leur base de données privée de taille inconnue. On peut par exemple imaginer que cela soit une API que l'on interroge en fournissant le pays d'origine d'un ingrédient, le pays où il est transformé et enfin le pays où il est consommé. Celle-ci pourrait nous renvoyer alors un fichier JSON constitué des principales routes commerciales empruntées par CMA-CGM pour le transport de cette marchandise, avec le ou les moyens de transport utilisé(s) et la distance parcourue. Couplé à la connaissance de l'intensité carbone par moyen de transport emprunté, on pourra alors déterminer l'empreinte carbone du transport du produit alimentaire mais aussi afficher à l'utilisateur l'itinéraire de son produit.

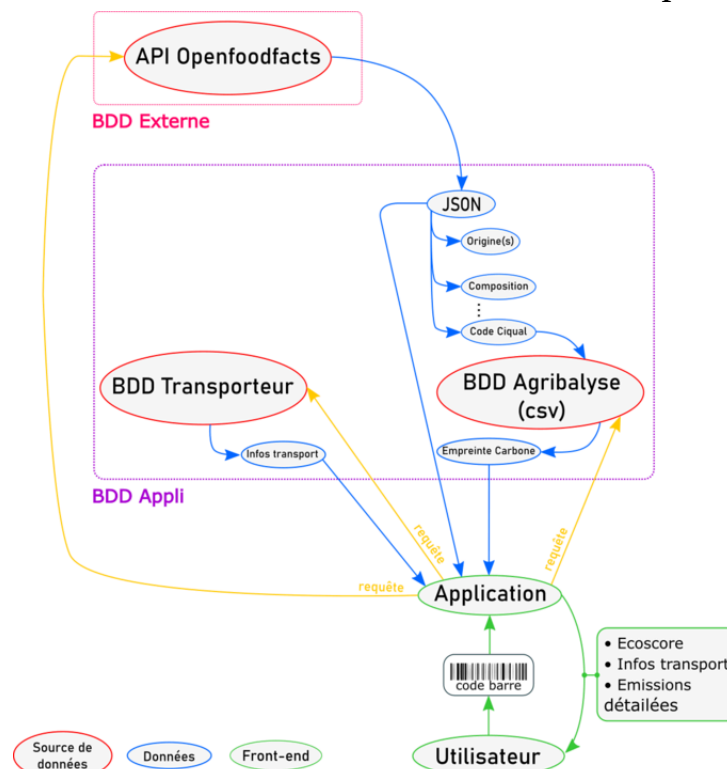


Figure 1 - Schéma de fonctionnement de FoodTrack

- Exactitude et exhaustivité des données

Plusieurs contrôles sont effectués. Nous avons mis en œuvre dans la démonstration des tests de cohérence pour vérifier par exemple que la somme des pourcentages des ingrédients d'un produit (renvoyés par l'API Open Food Facts) est bien égale à 100%. De plus, nous imaginons qu'il serait intéressant de conserver un historique des données utilisées pour calculer l'empreinte environnementale d'un produit, notamment pour des raisons juridiques. Ainsi, nous pourrions comparer les données récentes issues de l'API Open Food Facts à cet historique afin de vérifier l'absence de valeurs aberrantes.

Transformation des données

Les données provenant de l'API Open Food Facts subissent un nettoyage : seuls les champs pertinents du fichier JSON tels que la liste des ingrédients, leur proportion, leur origine, leur code CIQUAL et la masse du produit sont conservés. Ce sont ces données uniquement qui seront conservées dans l'historique.

Par ailleurs, dans la démonstration, seules 4 colonnes de la base de données AGRIBALYSE (le code AGRIBALYSE, le code CIQUAL, le nom de l'ingrédient et son facteur d'émission carbone) sont conservées et converties dans un fichier CSV avant d'être chargées dans la base de données PostgreSQL.

La figure 1 témoigne de la séquentialité du traitement des données : il est nécessaire d'extraire les données intéressantes du fichier JSON issu de l'API Open Food Facts avant de pouvoir adresser les requêtes aux bases de données AGRIBALYSE et transporteur. Par ailleurs, le calcul de l'empreinte carbone n'a lieu qu'une fois toutes ces requêtes effectuées. On ne relève aucun besoin d'enrichir les données extraites.

- Gestion des données manquantes et des erreurs

Nous avons assez rapidement remarqué que les données renvoyées par l'API Open Food Facts sont critiques. Ces dernières peuvent être incomplètes : c'est souvent le cas de l'origine des ingrédients. Nous avons choisi de communiquer cette absence de donnée à l'utilisateur, puisque c'est un bon indicateur du niveau de transparence du fabricant. Parfois, le code CIQUAL n'est pas systématiquement renseigné : nous pourrions alors développer une méthode de classification permettant d'assigner à l'ingrédient considéré un code CIQUAL vraisemblable. Enfin, certaines données renvoyées par cette API sont parfois incorrectes : lors d'un test avec une bouteille de Coca-Cola, nous avons constaté qu'un volume avait été renseigné dans le champ "masse du produit". Dès lors, nous pourrions peut-être directement signaler l'incohérence de cette valeur à l'ONG afin qu'elle corrige la donnée.

- Gestion des audits

Comme mentionné en fin de partie précédente, il serait intéressant de conserver les valeurs utilisées pour chaque calcul d'empreinte environnementale à des fins juridiques pendant une certaine durée à définir (probablement quelques mois).

Ainsi, si un fabricant nous interroge sur une donnée incohérente renvoyée par FoodTrack, nous pourrions être en mesure de la vérifier.

Chargement des données

Le système cible des données pour **FoodTrack** pourrait être un **entrepotage hybride** combinant les avantages des bases de données relationnelles pour les données structurées (PostgreSQL pour AGRIBALYSE) et une **base de données NoSQL** (comme MongoDB) pour gérer les informations provenant de l'API Open Food Facts (et éventuellement leur export complet).

- **PostgreSQL** convient parfaitement pour les données AGRIBALYSE. Ces données sont relationnelles, bien structurées, et les besoins de mise à jour sont peu fréquents (tous les 2 ans).
- **MongoDB** (ou un autre système NoSQL) serait bien adapté pour l'intégration de la base de données d'Open Food Facts (appels fréquents à l'API ou exports). MongoDB est conçu pour gérer des volumes de données massifs avec des structures de données flexibles comme des documents JSON.
- Interrogation d'API externe pour le transport : Plutôt que d'intégrer directement ces données dans une base de données, il serait plus efficace d'interroger en temps réel l'API des transporteurs pour récupérer les informations de trajet à la demande, en fonction des ingrédients et des lieux de production/consommation.

Dans le cadre de la démonstration, la base de données NoSQL n'a pas été implémentée : l'API Open Food Facts est directement interrogée et les données sont donc chargées en temps réel. Afin de s'assurer que le processus de chargement se déroule avec succès, nous surveillons les réponses de l'API afin de détecter d'éventuels problèmes ou incohérences.

Comme expliqué dans la partie précédente, les données de calcul sont conservées dans un historique (non implémenté dans la démonstration) à des fins d'audit. Nous relevons un besoin de certifier les données (ou les chiffrer) afin que personne ne puisse altérer les données entre le moment où l'utilisateur scanne le produit et celui où FoodTrack traite la demande puis renvoie le résultat. Nous nous sommes fixés comme seule contrainte un affichage rapide (de l'ordre de la seconde) de l'éco-score pour le consommateur.

Lien vers le repository Github de la démonstration :

https://github.com/Sami-bkz/foodtrack_demo