

18 JUIN 2023



CYBERCRAFT

PROJET CYBERCRAFT

WEB MINING

ANTONY CARRARD, KILLIAN VERVELLE

Table des matières

Démarrage du microservice	3
1 Contexte et objectifs	3
2 Données	3
2.1 Acquisition des performances des composants	4
2.2 Acquisition des prix des composants	5
2.3 Recherche du composant au meilleur prix.....	7
3 État de l'art.....	8
4 Conception / Cas d'utilisation / Architecture	9
4.1 Scrapping des données du site LDLC	10
4.2 Fusion des données des site UserBenchmark et LDLC	13
4.3 Scrapping des données du site Toppreise	15
5 Fonctionnalités.....	16
5.1 Sélection des exigences.....	16
5.2 Répartition du budget	16
5.3 Sélection des composants principaux et secondaires et vérification de la compatibilité	16
5.4 Scrapping des données des composants.....	17
6 Techniques, algorithmes et outils utilisés	18
6.1 Algorithme de sélection des composants	18
7 Planification, organisation et suivi répartition du travail (diagramme de Gantt)	19
8 Conclusion	20
9 Travail futur	20

Démarrage du microservice

Pour commencer, importez le projet depuis Github à l'aide de la commande suivante :

```
git clone https://github.com/killianvervelle/Cybercraft.git
```

Puis, lancez les commandes suivantes depuis le terminal de commande de votre IDE au sein du projet :

```
python3.8 -m venv venv
source ./venv/bin/activate
pip install --upgrade pip
pip install -r requirements.txt
cd CyberCraft/svc
uvicorn main:app --reload
```

Et enfin, rendez-vous à l'adresse suivante : <http://127.0.0.1:8000/>.

1 Contexte et objectifs

Il existe à ce jour plusieurs sites de configuration sur mesure d'ordinateurs et de benchmark de leurs composants. Les données sur lesquelles reposent ces services ne sont pas toujours mises à jour en temps réel ce qui va à l'encontre même de la pertinence de leurs recommandations aux utilisateurs. Notre microservice permettra de répondre aux mêmes besoins utilisateur de manière centralisée et au travers d'une interface utilisateur simplifiée, tout en garantissant l'exactitude et l'exhaustivité des données renvoyées à l'utilisateur (compatibilité technique, prix actuels...).

L'objectif du projet est de développer un microservice capable, en toute autonomie, de configurer un ordinateur :

- Répondant aux contraintes utilisateurs ;
- Reposant sur les dernières technologies de composants disponibles sur le marché et au meilleur prix ;
- Proposant les meilleurs prix des composants du jour en Suisse ;
- Respectant les normes de comptabilité technique.

Le but est de proposer un configurateur d'ordinateur rapide et paramétrable, sans dépasser le budget de l'utilisateur.

Quant au périmètre du projet, le microservice portera principalement sur la configuration d'ordinateurs prévus à des utilisations gourmandes en ressources telles que le gaming, la modélisation et rendu 3d, le minage de cryptos...

2 Données

Nous avons utilisé des données provenant de 3 sites différents. Chacun de ces sites est utilisé pour une partie spécifique du projet.

Avec notre configurateur, nous cherchons à obtenir les meilleures performances possibles pour les composants proposés sans dépasser un budget imposé. Ainsi, nous avons besoin de connaître les performances ainsi que les prix de tous les composants d'ordinateurs actuels.

Pour cela, nous avons besoin d'une base de données contenant ces informations. Générer cette base de données au moment de la requête de l'utilisateur prendrait bien trop de temps dû au scrapping des données. Ainsi, nous devons créer cette base de données en dur.

2.1 Acquisition des performances des composants

Le premier site que nous avons utilisé est le site [UserBenchmark](#). Ce site propose aux utilisateurs de pouvoir effectuer un test de performances sur leur ordinateur. Les performances sont ensuite affichées pour chaque composant, et comparées aux performances des autres utilisateurs. Chaque composant est ensuite listé et un indice de performance est estimé en fonction de la moyenne des performances obtenues pour les utilisateurs qui en sont équipés. C'est pourquoi nous avons choisi ce site pour obtenir les données des composants actuels, car les composants sont testés en continu par des utilisateurs du monde entier.

Un autre avantage de ce site est qu'il propose [une page développeur](#). Des fichiers CSV peuvent être téléchargés avec les performances obtenues pour chacun des composants suivants :

CSV

Data Files







CPU (1,406)	
GPU (701)	
SSD (1,071)	
HDD (1,015)	
RAM (115)	
USB (639)	

Figure 1 - Fichiers CSV pouvant être téléchargés directement depuis la page développeur du site

Les catégories suivantes sont présentes dans les fichiers de données :

Fields

Type	Part Number	Brand	Model	Rank	Benchmark	Samples	URL
enum (CPU GPU SSD HDD USB RAM)	string	string	string	int	float	int	string

Figure 2 - Catégories des fichiers de données

Le scrapping n'est donc pas nécessaire pour ce site. Cependant, les données devront être prétraitées pour être utilisables. Cette partie sera abordée dans la suite du traitement des données.

2.2 Acquisition des prix des composants

Le second site utilisé est un site proposant d'effectuer sa configuration d'ordinateur, puis d'acheter les composants sélectionnés. Il s'agit du site de vente en ligne [LDLC](#).

L'avantage de ce site est qu'il propose tous les composants nécessaires à la conception d'un ordinateur sur une seule page, ce qui évite de devoir recharger un nombre important de pages web pour pouvoir obtenir toutes les données que nous recherchons.

Ce site se présente de la manière suivante, sous la forme d'un configurateur de PC.

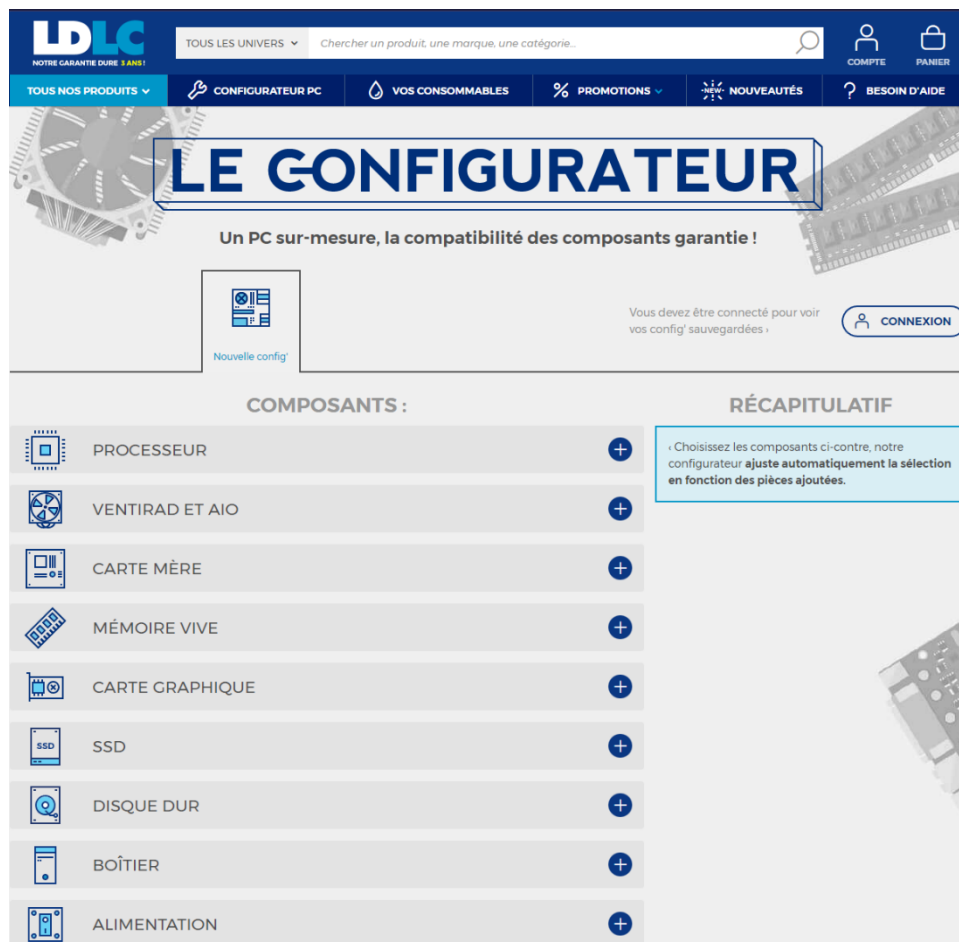


Figure 3 - Page d'accueil du configurateur de PC du site LDLC

L'utilisateur peut ensuite cliquer sur une catégorie, et choisir le composant qu'il l'intéresse. Il est aussi possible de filtrer les produits recherchés.

DÉSIGNATION	NOTE	PRIX	DISPO
Intel Core i5-13600KF (3.5 GHz / 5.1 GHz)	★★★★★	CHF 327 ⁹⁵	●
AMD Ryzen 7 5800X (3.8 GHz / 4.7 GHz)	★★★★★	CHF 237 ⁹⁵	●
AMD Ryzen 5 5600X Wraith Stealth (3.7 GHz / 4.6 GHz)	★★★★★	CHF 161 ⁹⁵	●
AMD Ryzen 5 5500 Wraith Stealth (3.6 GHz / 4.2 GHz)	★★★★★	CHF 110 ⁹⁵	●
Intel Core i7-12700KF (3.6 GHz / 5.0 GHz)	★★★★★	CHF 320 ⁹⁵	●
Intel Core i5-12400F (2.5 GHz / 4.4 GHz)	★★★★★	CHF 177 ⁹⁵	●
AMD Ryzen 5 3600 (3.6 GHz / 4.2 GHz)	★★★★★	CHF 85 ⁹⁵	●
Intel Core i5-12600KF (3.7 GHz / 4.9 GHz)	★★★★★	CHF 247 ⁹⁵	●
Intel Core i7-13700KF (3.4 GHz / 5.4 GHz)	★★★★★	CHF 443 ⁹⁵	●
AMD Ryzen 5 7600X (4.7 GHz / 5.3 GHz)	★★★★★	CHF 271 ⁹⁵	●
Intel Core i9-13900K (3.0 GHz / 5.8 GHz)	★★★★★	CHF 645 ⁹⁵	●
AMD Ryzen 5 5600 Wraith Stealth (3.5 GHz / 4.4 GHz)	★★★★★	CHF 156 ⁹⁵	●

[Comparer](#)

Intel Core i5-13600KF (3.5 GHz / 5.1 GHz) -
 Processeur 14-Core (6 Performance-Cores + 8 Efficient-Cores) 20-Threads Socket 1700
 Cache L3 24 Mo 0.010 micron (version boîte sans ventilateur - garantie Intel 3 ans)

Figure 4 - Pop-up de sélection d'un processeur

Parmi ces données, nous avons besoin de la désignation du produit, ainsi que son prix, pour pouvoir fusionner ces données avec les données des performances des composants.

Des données intéressantes se trouvent également dans la description des produits, qui apparaît à droite du pop-up lorsque l'élément en question est survolé par la souris de l'utilisateur. Cette description contient des données qui pourront par la suite être utilisées pour filtrer les composants voulus par l'utilisateur, ainsi que pour vérifier la compatibilité du composant avec les autres. Dans ce cas, des données intéressantes seraient le nombre de cœurs d'un processeur, ainsi que son type de socket, pour vérifier la compatibilité avec la carte mère.

Nous avons récupéré les données des catégories suivantes :

COMPOSANTS :

- PROCESSEUR
- VENTIRAD ET AIO
- CARTE MÈRE
- MÉMOIRE VIVE
- CARTE GRAPHIQUE
- SSD
- DISQUE DUR
- BOÎTIER
- ALIMENTATION

Figure 5 - Catégories desquelles nous avons récupéré les données.

2.3 Recherche du composant au meilleur prix

Le dernier site que nous avons utilisé est le site Suisse de comparaison [Toppreise](#), qui propose un site de vente en ligne Suisse proposant le composant recherché au meilleur prix disponible. Une fois la liste des composants établie par notre configurateur, c'est sur ce site que nous allons chercher le composant du moment le moins cher possible.

The screenshot displays the Toppreise.ch website interface for comparing Intel Core i7-13700K processors. The page features a navigation bar with tabs for 'Infos produit', '24 offres', 'Fiche technique', 'Variantes de produit', '4 Enchères', 'Autres produits', and 'Évaluations'. Below the navigation bar, there are filters for 'Envoi (24)' and 'Retrait (30)', and a search bar with the text 'Recherche de produit'. The main content area lists several offers for the Intel Core i7-13700K processor, each with a price, shipping cost, and a rating. The offers are sorted by price, with the lowest price being CHF 399.50 from reichelt. The highest price is CHF 412.00 from BRACK.CH. The page also includes a 'Prix d'envoi le moins cher' badge and a 'Prix du produit' badge.

Offre	Prix	Prix d'expédition	Notes
Intel - Core i7-13700K (16C, 3.40GHz, 30MB, boxed) (BX8071513700K)	CHF 402.75	plus expédition: 0.00	Prix d'envoi le moins cher
Intel Core i7-13700K (16C, 3.40GHz, 30MB, boxed) (BX8071513700K)	CHF 402.80	plus expédition: 0.00	
Solidigm CPU Intel Core i7-13700K / LGA1700 / Box 16 Cores / 24 Threads / 30M Cache (BX8071513700K)	CHF 404.25	plus expédition: 0.00	
Intel CPU Intel Core i7-13700K / LGA1700 / Box 16 Cores / 24 Threads / 30M Cache (BX8071513700K)	CHF 404.30	plus expédition: 0.00	
Intel CPU i7-13700K 2.5 GHz, Prozessorfamilie: Intel Core i7 (13xxx), Anzahl Prozessorkerne: 16, Arbeitsspeicher Geschwindigkeit: 5600... (BX8071513700K)	CHF 404.35	plus expédition: 0.00	
INTEL BX8071513700K - Intel Core i7-13700K, 3.40GHz, boxed ohne Kühler, 1700 Marchand d'Allemagne - les prix sont indiqués avec TVA Suisse et dédouanement	CHF 399.50	plus expédition: 9.50	Prix le plus bas
INTEL Core i7-13700K (LGA 1700, 2.5 GHz) (BX8071513700K)	CHF 411.70	plus expédition: 0.00	
Intel Core i7-13700K BX8071513700K	CHF 412.00	plus expédition: 0.00	
Intel CPU i7-13700K 2.5 GHz, Prozessorfamilie: Intel Core i7 (13xxx), Anzahl Prozessorkerne: 16, Arbeitsspeicher Geschwindigkeit:.... (BX8071513700K)	CHF 412.00	plus expédition: 0.00	

Figure 6 - Un exemple de comparatif pour un processeur de modèle Core i7-13700K

C'est ensuite le lien vers le site de vente en ligne proposant le composant le moins cher du moment qui sera donné au client via notre configurateur.

Nous pouvons ici noter que les prix proposés sur le site [LDLC](#) sur lequel nous avons obtenus tous les prix des composants peuvent différer légèrement des prix que nous retrouvons sur [Toppreise](#). Cependant, nous avons préféré utiliser le site [LDLC](#) pour récupérer tous les prix des composants, car cela nous permet de récupérer le prix de milliers de composants sur une seule page web, contrairement à [Toppreise](#) qui nécessiterait un temps de recherche beaucoup plus long pour obtenir la même quantité de données. Ainsi, bien que nous puissions avoir quelques imprécisions sur les prix de références, ils demeurent suffisamment proche pour effectuer le choix en amont des meilleurs composants.

3 État de l'art

Les configurateurs de PC existants offrent généralement une interface web permettant aux utilisateurs de choisir les composants en fonction de certains filtres et de leur prix. Les compatibilités entre les composants sont vérifiées dynamiquement, de sorte que lorsque l'utilisateur sélectionne un composant, seuls les composants compatibles restent disponibles pour la sélection suivante.

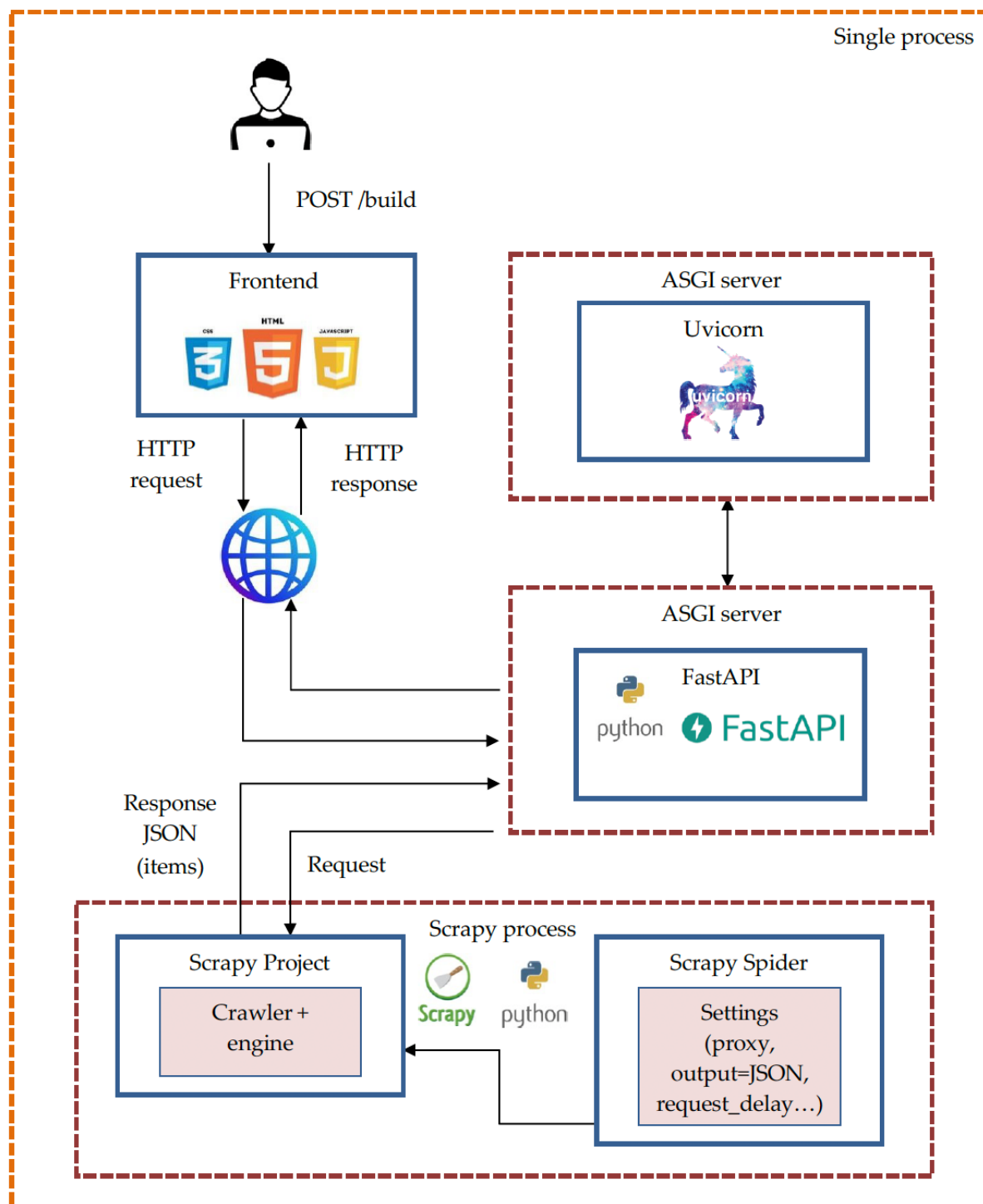
Ces configurateurs de PC sont souvent intégrés directement sur des sites de vente en ligne, tels que [LDLC](#), ou présents sur des sites spécialisés proposant des configurations préexistantes et un sélecteur de composants, à l'instar de [PCPartPicker](#).

Bien que ces outils fonctionnent bien, l'utilisateur doit ensuite acheter les composants sur le site de vente en question ou rechercher chaque composant sur d'autres sites de vente en ligne. De plus, la sélection des composants nécessite un certain niveau de connaissances informatiques de la part de l'utilisateur, car il doit être en mesure d'estimer si les composants qu'il a sélectionnés répondront à ses exigences de performances par rapport à son budget.

Notre configurateur vise à résoudre ces deux problèmes. Tout d'abord, les composants sélectionnés sont déjà au meilleur prix du marché actuel grâce au site de comparaison [Toppreise](#). L'utilisateur n'a qu'à cliquer sur le lien proposé pour acheter le composant. Ensuite, la liste des composants est générée de manière optimale et automatique en fonction du budget de l'utilisateur, ce qui signifie qu'il n'est pas nécessaire pour lui d'avoir des connaissances informatiques spécifiques. Cependant, grâce aux filtres supplémentaires que nous avons ajoutés, l'utilisateur peut quand même spécifier certaines préférences pour certains composants.

Ainsi, notre configurateur offre une solution pratique en proposant une sélection optimisée de composants, des prix compétitifs et une interface simplifiée pour les utilisateurs, même sans expertise informatique.

4 Conception / Cas d'utilisation / Architecture



Notre microservice de Scraping Web, développé avec FASTAPI et Scrapy, permet d'extraire des données à partir de sites web de manière efficace et rapide, en fonction d'une requête HTTP utilisateur contenant des exigences. En combinant les fonctionnalités robustes de FASTAPI et les capacités de scraping de Scrapy, ce microservice offre une solution complète automatisant le processus de récupération de données en ligne.

Le processus de configuration d'un pc s'est construit en deux étapes. Tout d'abord, nous avons développé des scrappeurs de données qui collectent les informations nécessaires pour construire notre base de données de références concernant le choix des composants. Ensuite, nous avons effectué un pré-traitement de ces données afin de les exploiter de manière efficace.

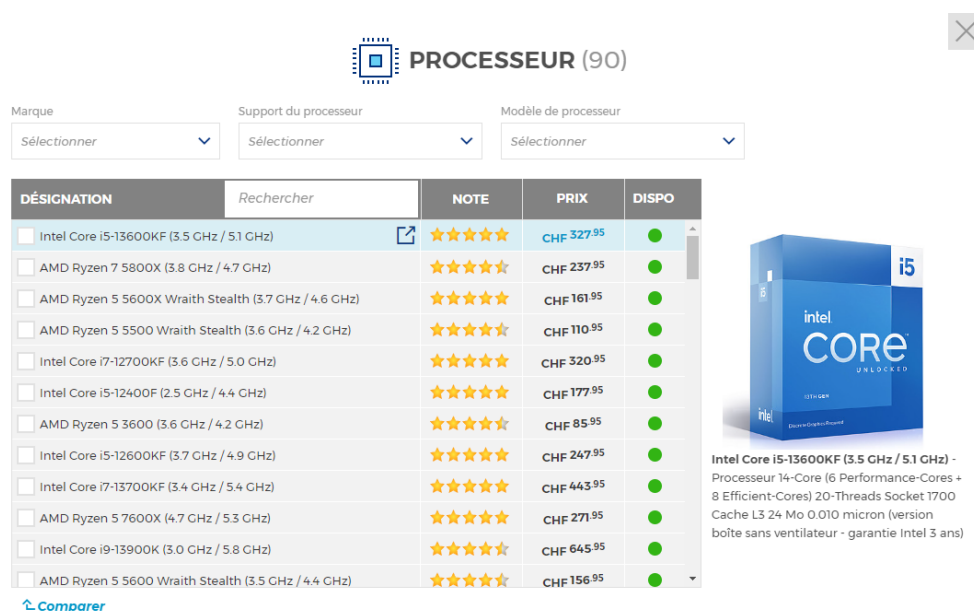
Dans la seconde étape, lorsqu'un utilisateur effectue une requête sur notre page web, un algorithme d'optimisation est lancé. Cet algorithme analyse la requête de l'utilisateur et exploite les données disponibles dans notre base de données pour identifier les meilleurs composants répondant à ses besoins spécifiques. Il tient compte de plusieurs critères essentiels tels que le budget alloué, les performances recherchées, les préférences personnelles de l'utilisateur... Grâce à cette approche, nous sommes en mesure de proposer des configurations de PC optimisées et sur mesure pour chaque utilisateur.

4.1 Scrapping des données du site LDLC

Comme vu précédemment dans la section Données, nous n'avons pas eu besoin de scraper les données du site [UserBenchmark](#) duquel nous avons récupéré les performances des composants. Cependant, nous avons scrapé les données du site [LDLC](#) pour obtenir les prix et quelques informations sur les composants.

Nous avons utilisé le framework Selenium, qui permet d'effectuer du scrapping sur des pages web dynamique ainsi que d'effectuer des tests utilisateurs. Nous avons choisi ce framework pour cette partie car elle permet d'obtenir des données qui sont chargées dynamiquement sur la page web, comme c'est le cas ici.

En effet, lorsqu'un clic est effectué sur une des catégories de composants, les composants listés sont chargés dynamiquement. Il est alors nécessaire d'attendre la fin de la requête, puis d'aller chercher les données présentes dans la liste.



PROCESSEUR (90)

Marque: Sélectionner | Support du processeur: Sélectionner | Modèle de processeur: Sélectionner

DÉSIGNATION	Rechercher	NOTE	PRIX	DISPO
<input type="checkbox"/> Intel Core i5-13600KF (3.5 GHz / 5.1 GHz)	🔗	★★★★★	CHF 327 ⁹⁵	●
<input type="checkbox"/> AMD Ryzen 7 5800X (3.8 GHz / 4.7 GHz)		★★★★★	CHF 237 ⁹⁵	●
<input type="checkbox"/> AMD Ryzen 5 5600X Wraith Stealth (3.7 GHz / 4.6 GHz)		★★★★★	CHF 161 ⁹⁵	●
<input type="checkbox"/> AMD Ryzen 5 5500 Wraith Stealth (3.6 GHz / 4.2 GHz)		★★★★★	CHF 110 ⁹⁵	●
<input type="checkbox"/> Intel Core i7-12700KF (3.6 GHz / 5.0 GHz)		★★★★★	CHF 320 ⁹⁵	●
<input type="checkbox"/> Intel Core i5-12400F (2.5 GHz / 4.4 GHz)		★★★★★	CHF 177 ⁹⁵	●
<input type="checkbox"/> AMD Ryzen 5 3600 (3.6 GHz / 4.2 GHz)		★★★★★	CHF 85 ⁹⁵	●
<input type="checkbox"/> Intel Core i5-12600KF (3.7 GHz / 4.9 GHz)		★★★★★	CHF 247 ⁹⁵	●
<input type="checkbox"/> Intel Core i7-13700KF (3.4 GHz / 5.4 GHz)		★★★★★	CHF 443 ⁹⁵	●
<input type="checkbox"/> AMD Ryzen 5 7600X (4.7 GHz / 5.3 GHz)		★★★★★	CHF 271 ⁹⁵	●
<input type="checkbox"/> Intel Core i9-13900K (3.0 GHz / 5.8 GHz)		★★★★★	CHF 645 ⁹⁵	●
<input type="checkbox"/> AMD Ryzen 5 5600 Wraith Stealth (3.5 GHz / 4.4 GHz)		★★★★★	CHF 156 ⁹⁵	●

[↑ Comparer](#)

Intel Core i5-13600KF (3.5 GHz / 5.1 GHz) -
 Processeur 14-Core (6 Performance-Cores + 8 Efficient-Cores) 20-Threads Socket 1700
 Cache L3 24 Mo 0.010 micron (version boîte sans ventilateur - garantie Intel 3 ans)

Figure 7 - La liste des composants, s'affichant sous forme d'une table

La description du produit, à droite du pop-up, ne s'affiche que si le client survole un élément en particulier avec la souris. Si l'on souhaite récupérer toutes les descriptions, il est nécessaire de survoler les éléments de toute la table.

Finalement, lorsqu'il y a beaucoup d'éléments présents dans une catégories, comme ici plus de 1000, la table les composants ne sont pas tous chargés dans la table. Il est donc nécessaire de scroller dans la table, et d'attendre régulièrement que les prochaines données aient chargé.



Figure 8 - Les mémoires vives, qui sont en train de charger suite à un scrolling utilisateur

Le scrapper va alors parcourir toute la table, et répéter l'opérations pour chaque composants PC. Les données sont ensuite écrites dans des fichiers CSV, un par type de composant, avec les attributs « designation », « price » et « description ».

Voici le diagramme de flux effectué par le scrapper utilisant Selenium :

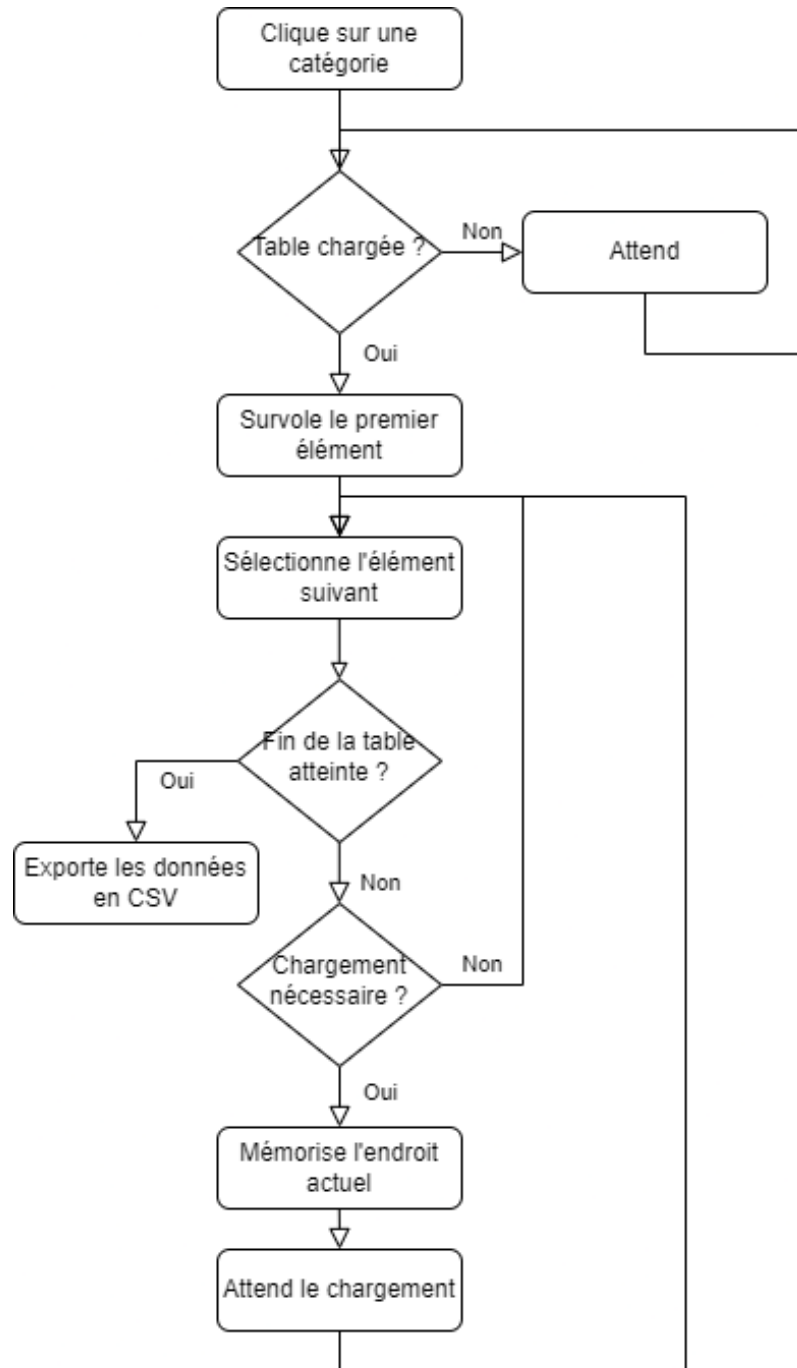


Figure 9 - Diagramme de flux effectué par le scrapper utilisant Selenium

4.2 Fusion des données des site UserBenchmark et LDLC

Afin d'effectuer la mise en commun des données des deux sites internet et de pouvoir par la suite faire le choix des composants en fonction de leur prix et de leur performance, il est nécessaire de fusionner ces données.

Dans le fichier CSV provenant du site UserBenchmark, nous avons les attributs suivants :

Fields

Type	Part Number	Brand	Model	Rank	Benchmark	Samples	URL
enum (CPU GPU SSD HDD USB RAM)	string	string	string	int	float	int	string

Figure 10 - Attributs des fichiers CSV provenant du site UserBenchmark

Sur les fichiers CSV que nous avons établi à la suite du scrapping du site LDLC, nous avons les attributs suivants :

designation	price	description
string	float	string

Afin d'avoir un champ commun pour pouvoir effectuer la fusion de données, nous avons récupéré le modèle des composants à partir de la colonne « designation ».

Les modèles des composants ont été extraits à l'aide d'expressions régulières. Par exemple, voici les expressions régulières qui ont été utilisées pour extraire les processeurs de marque Intel et AMD de la colonne « designation » :

```
# Get the model of the cpu
re_core = r'\b(Core\s+i\d-\d+[a-zA-Z]*)\b'
re_ryzen = r'\b(Ryzen\s+[0-9]\s+\w*\s*\d+\w*)\b'
self.cpu['model'] = self.cpu['model'] =
self.cpu['designation'].str.extract(re_core) \
.combine_first(self.cpu['designation'].str.extract(re_ryzen))
```

Voici également les expressions régulières qui ont été utilisées pour extraire le modèle des cartes graphiques :

```
# Get the model of the gpu
re_model_nvidia = r'\b([GR]TX*\s+\d+\s*T*i*)\b'
re_model_amd = r'\b(RX\s+\d+\s*X*T*X*)\b'
self.gpu['model'] = self.gpu['description'].str.extract(re_model_nvidia) \
.combine_first(self.gpu['description'].str.extract(re_model_amd))
```

Les disques SSD ainsi que les barrettes de RAM sont parfois disponibles avec le même nom de modèle mais avec des capacités différentes. Il est donc nécessaire d'extraire la capacité de la « designation » et de l'ajouter dans une nouvelle colonne « memory ».

Voici une fonction qui extrait la mémoire d'une cellule donnée, et qui multiplie le résultat par 1000 si la mémoire est notée sous forme de téraoctet.

```
def get_memory(self, cell:str, re_memory:str, tb_to_gb:int=1000):
    match = re.search(re_memory, cell)
    if match:
        value = int(match.group(1))
        unit = match.group(2)
        if unit == 'T':
            value *= tb_to_gb
        return value
    else:
        return None
```

Avec l'expression régulière suivante passée en paramètre :

```
re_memory = r'\s*(\d+)\s*(T|G)[Bo].*'
```

Pour la mémoire vive, étant donné que le numéro de série est présent dans les deux fichiers de données, il s'est avéré plus simple d'effectuer la fusion directement à partir de ce numéro de série. L'expression régulière pour l'extraire est la suivante :

```
# Get the part number from the ram
re_part_number = r'\s+-\s+\b([A-Z0-9\\\/-]{10,})'
self.ram['part number'] = self.ram['description'].str.extract(re_part_number)
```

Pour les disques SSD, il s'est avéré difficile d'extraire la désignation, car elle n'a pas de forme particulière contrairement aux processeurs ou cartes graphiques. Une méthode a donc été d'extraire tout ce qui n'appartenait pas au modèle du SSD dans la « designation », à savoir une liste de marque :

```
brands_names = ['Samsung', 'Crucial', 'Kingston', 'Western Digital', 'Fox Spirit', 'Seagate', 'Textorm', 'Intel', 'Corsair', 'LDLC']
```

Et une liste de mots utilisés pour désigner des caractéristiques de la mémoire :

```
terms_to_remove = ['SSD', 'M.2', 'PCIe', 'NVMe', '2280', 'NAND', '3D', 'WD']
```

En enlevant ces termes, ainsi que l'information de la quantité de mémoire, nous nous retrouvons plus qu'avec le nom de modèle, qui peut être utilisé pour la fusion de données.

Une fois ces informations obtenues, les deux fichiers prétraités ont été fusionnés en fonction du modèle et de la marque pour les processeurs et les cartes graphiques, du numéro de série pour la mémoire vive, ainsi que le modèle, la marque et la capacité de la mémoire pour les disques SSD.

Les informations à double, tel que le prix des cartes graphiques du même modèle, ont été moyennées lors de la fusion.

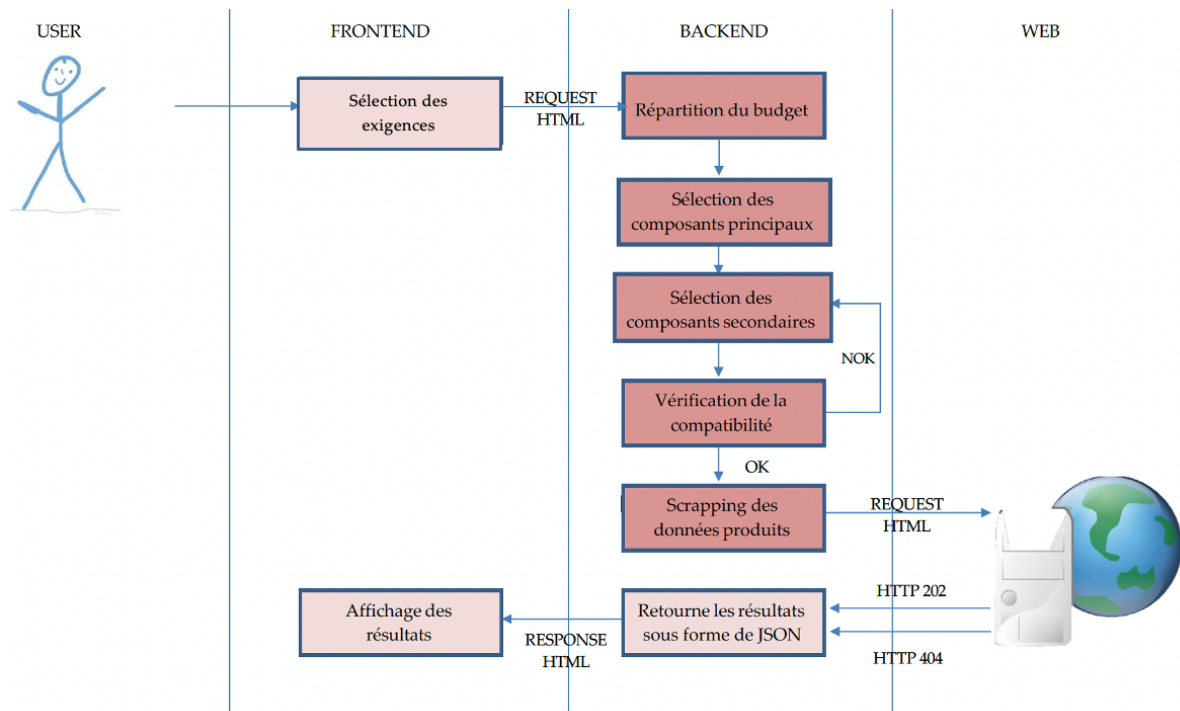
4.3 Scrapping des données du site Toppreise

[Toppreise.ch](https://toppreise.ch) collecte des informations sur les produits électroniques à partir de diverses sources, y compris les détaillants en ligne, les sites web des fabricants et les bases de données de produits. Ils utilisent également des techniques de scraping web et travaille en partenariat avec des fournisseurs pour obtenir ces données.

Les informations collectées sont ensuite organisées et structurées dans une base de données centrale. Cela permet à [Toppreise.ch](https://toppreise.ch) d'accéder rapidement aux informations sur les produits et de les comparer de manière efficace. Ainsi, il nous est très facile de rechercher des produits spécifiques et [Toppreise.ch](https://toppreise.ch) nous fournira une liste de résultats avec les prix des différents détaillants en ligne qui proposent ces produits.

Notre scraper prend en entrée une liste des désignations des produits sélectionnés, interroge successivement [Toppreise.ch](https://toppreise.ch) et en extrait les meilleures offres fournisseurs ainsi que les URLs de leur page web. Ces données sont ensuite formatées et renvoyées, sous la forme d'une réponse à la requête HTML, dans le frontend.

5 Fonctionnalités



Le microservice est composé de plusieurs fonctions, permettant de traduire les exigences de l'utilisateur en paramètres, sélectionner les composants appropriés, vérifier leur compatibilité technique et enfin extraire les données des composants ciblés. Voici une description détaillée de chaque fonction et de son rôle tout au long du processus :

5.1 Sélection des exigences

L'utilisateur peut spécifier des exigences dans sa requête afin d'orienter le microservice vers des configurations plus spécifiques. Ces exigences sont traduites en paramètres et transmises au backend du microservice. Les fichiers de données sont ensuite filtrés à partir de ces paramètres, ce qui permet au configurateur de ne considérer que les composants d'intérêts.

5.2 Répartition du budget

Le paramètre le plus restrictif est le budget. Ainsi, il subira une répartition selon des pourcentages repris de statistiques fournis par plusieurs sites, notamment [UserBenchmark](#). Cette répartition est de 70% pour les composants principaux tel que le CPU, le GPU, la RAM et le SSD. Les répartitions exactes de chacun de ces composants sont attribuées lors de l'étape d'optimisation. Le 30% du budget restant est attribué à la sélection des composants secondaire.

5.3 Sélection des composants principaux et secondaires et vérification de la compatibilité

La sélection des composants se fait en cascade, en débutant par les composants les plus impactant sur la performance du système. Ils seront nommés les composants « principaux ». Ils ne répondent à aucune exigence de compatibilité et sont donc sélectionnés indépendamment. Le configurateur sélectionne ces composants, à partir des paramètres d'entrées et de leur score benchmark. Le configurateur va ensuite sélectionner les composants avec les performances maximales qui peuvent être permises par le budget du client.

Une fois la sélection des composants principaux terminée, le configurateur se tourne vers les composants secondaires, qui, quant à eux, sont concernés par des normes de compatibilité. Nous les avons extraites à partir du site www.ldlc.ch et les avons traduites sous la forme de conditions au sein de notre code afin de tester pour chacun des composants secondaires, leur compatibilité avec l'ensemble des composants déjà sélectionnés. Dans le cas où un composant s'avère incompatible, le configurateur prend tout simplement le composant suivant, et ce jusqu'à la satisfaction de toutes les règles de compatibilité. A savoir que la sélection des composants secondaires se base sur leur prix respectif, et non leur score benchmark, cette donnée n'étant pas toujours disponible.

5.4 Scrapping des données des composants

Et enfin, une fois la liste des composants établie, le configurateur démarre un processus de scrapping à partir d'une liste contenant les désignations des composants sélectionnés, sur le site www.toppreise.ch et récupère les dernières offres fournisseurs disponibles sur le marché. La réponse à la requête utilisateur est alors générée et renvoyée dans le frontend. L'utilisateur peut alors consulter la liste des composants sélectionnés et directement composer son panier d'achat en cliquant sur les URLs des produits.

6 Techniques, algorithmes et outils utilisés

6.1 Algorithme de sélection des composants

Afin de sélectionner les composants principaux à transmettre au client, nous avons utilisé un algorithme d'optimisation greedy de type branch-and-bound.

Le problème posé est le suivant : Nous voulons obtenir, pour une certaine contrainte de prix, les composants qui peuvent fournir un maximum de performances. Ces composants sont des composants directement responsables des performances de l'ordinateur, à savoir le processeur, la carte graphique, la mémoire vive ainsi que le disque SSD.

Pour ces 4 composants, une partie du budget total est alloué. Ici, nous avons choisi 70 %, car la majorité du budget total va dans ces composants qui sont principaux.

Ainsi, nous cherchons à maximiser :

$$performance_{CPU} + performance_{GPU} + performance_{RAM} + performance_{SSD}$$

Avec la contrainte de prix suivante :

$$budget_{CPU} + budget_{GPU} + budget_{RAM} + budget_{SSD} \leq budget_{alloué}$$

Les performances sont issues du site [UserBenchmark](#). Afin de donner une priorité aux mémoires vives et aux disques SSD qui ont une plus grande capacité, les performances de ces composants sont multipliées par le logarithme naturel de leur capacité. Un logarithme est ici utilisé pour ne pas trop pénaliser les composants qui seraient performants mais qui auraient des plus petites capacités. Les performances de chaque catégorie sont ensuite normalisées entre 0 et 1.

Chaque fichier de données est ensuite trié du composant le plus au moins performant.

Le pseudo code de l'algorithme est le suivant :

```
cout_actuel = cout_cpu + cout_gpu + cout_ram + cout_ssd
while cout_actuel > budget
    // Filtre les composants + cher et - performants que ceux actuels
    filter(cpu, gpu, ram, ssd)
    // Sélectionne le prochain composant directement - performant
    select_next_componant(cpu, gpu, ram, ssd)
    // Calcule à nouveau le coût actuel des composants
    cout_actuel = cout_cpu + cout_gpu + cout_ram + cout_ssd
perf_finale = perf_cpu + perf_gpu + perf_ram + perf_ssd
```

Avec `cpu`, `gpu`, `ram` et `ssd` qui sont des dataframes contenant les colonnes de performance et de coût. Étant donné que les dataframes sont triés, les composants actuellement sélectionnés sont automatiquement ceux de la première ligne de chaque dataframe.

Lors de la sélection du prochain composant directement moins performant, la prochaine catégorie dont le composant est le plus puissant est sélectionnée, et le premier composant du dataframe correspondant est supprimé. Le prochain composant est alors automatiquement sélectionné.

Une vérification est également effectuée au préalable, afin de s'assurer que le budget demandé permette de proposer une configuration possible de composants.

7 Planification, organisation et suivi répartition du travail (diagramme de Gantt)

Killian a travaillé sur la conception du microservice dans FastApi, front et backend, le scrapping des données du site [Toppreise](#), ainsi que le scrapping de certaines données du site [LDLC](#). Antony a travaillé sur la partie du scrapping du site [LDLC](#), le pré-traitement et la fusion des données avec le site [UserBenchmark](#), ainsi que sur l'algorithme de sélection des composants. Nous avons tous deux rédigé le rapport.

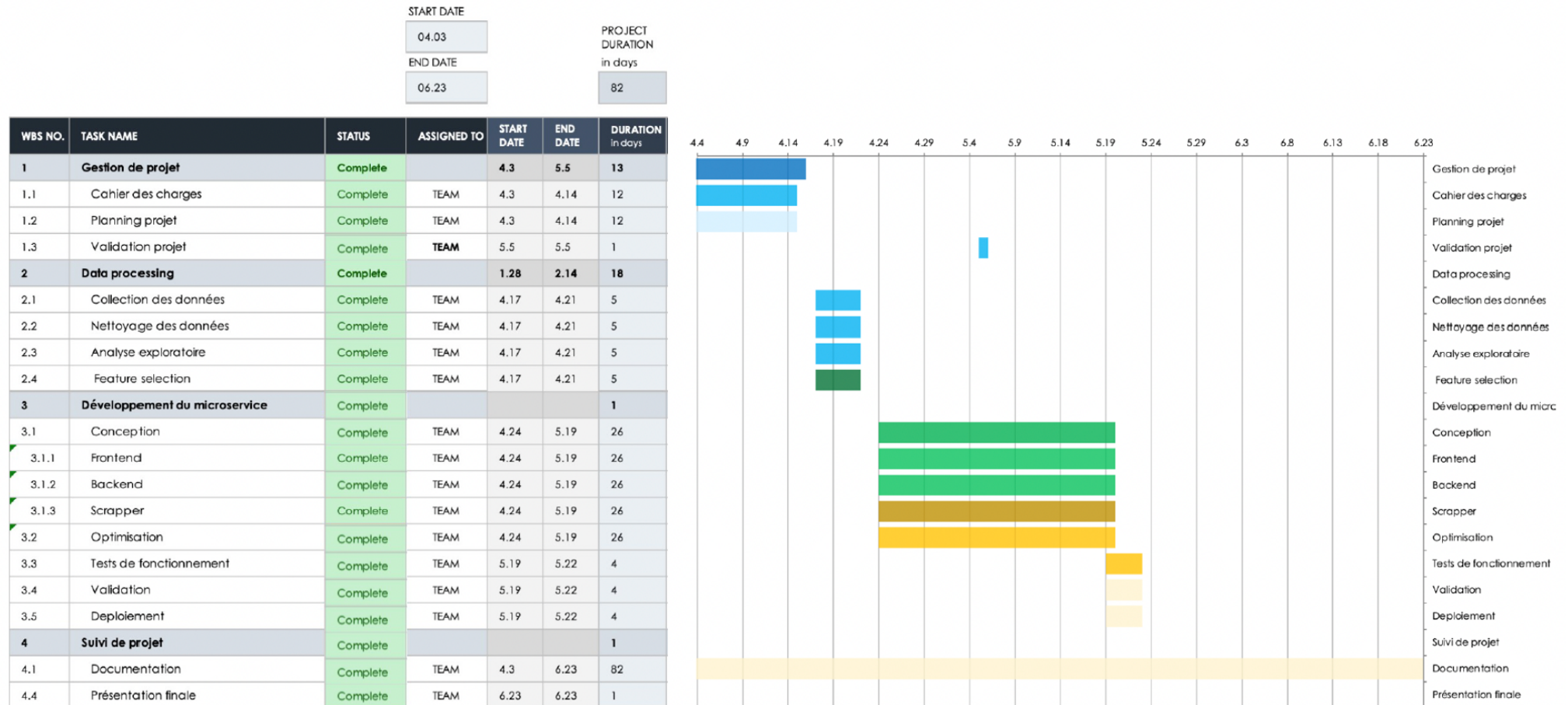


Figure 11 - Diagramme de Gantt du projet

8 Conclusion

Notre objectif principal était de créer un microservice capable de proposer aux utilisateurs une configuration d'ordinateur prête à l'achat, avec les meilleurs prix du moment et les meilleures performances possibles en fonction de leurs exigences. Nous avons réussi à réaliser un produit complet qui englobe l'acquisition des données, le pré-traitement, la sélection des composants et la proposition d'options aux utilisateurs à des prix optimaux.

Pour atteindre cet objectif, nous avons utilisé des outils de traitement de données tels que le framework Pandas et les expressions régulières. Le scrapping des données a été réalisé à l'aide de frameworks couramment utilisés tels que Scrapy, Selenium et BeautifulSoup. Nous avons également utilisé des outils web, notamment FastAPI, pour la conception du FrontEnd. La programmation dynamique a été appliquée pour sélectionner les composants idéaux à proposer aux utilisateurs.

Le résultat final est un configurateur de PC conçu pour les utilisateurs qui recherchent un outil simple à utiliser. Il leur permet d'obtenir des composants adaptés à leurs besoins, même sans connaissances préalables en informatique.

9 Travail futur

Bien que notre produit soit fonctionnel et abouti, il est toujours possible de le perfectionner en apportant quelques améliorations supplémentaires :

- 1) Ajouter des filtres supplémentaires dans la sélection des composants afin d'offrir aux utilisateurs une plus grande flexibilité pour personnaliser leur configuration.
- 2) Inclure des images des composants choisis : L'ajout d'images des composants sélectionnés aidera les utilisateurs à visualiser leur configuration et à avoir une meilleure idée de l'apparence finale de leur ordinateur. Nous pouvons obtenir ces images auprès des fabricants ou créer des illustrations personnalisées pour chaque composant.
- 3) Estimation des performances du matériel : L'idée d'estimer les performances du matériel sélectionné en se basant sur les données du site UserBenchmark est intéressante. Nous pouvons intégrer une fonctionnalité qui récupère les données pertinentes de UserBenchmark pour les composants sélectionnés par l'utilisateur et génère une estimation des performances globales de l'ordinateur. Cela permettra aux utilisateurs d'avoir une idée plus concrète des performances attendues.