BigData

Compétences et évaluation

E07 - Développer des composants d'interface avec d'autres applications

E08 - Installer et configurer une architecture de stockage et de traitement de données distribuées volumineuses

E09 - Maîtriser un Framework de construction d'applications distribuées

E10 - Développer des requêtes SQL et NoSQL pour traiter des données volumineuses

Data Lake

Points	Critère observable
0	Pas de data lake disponible
2	Le data lake est en place, les données initiales sont accessibles en lecture (sur le filesystem local ou sur une architecture de stockage distribuée locale ou cloud)
3	Le data lake est enrichi (lecture et écriture) avec des données supplémentaires (issues d'un pré-traitement ou de sources externes)
5	Le data lake est sur une architecture de stockage distribuée type HDFS (lecture et écriture)

Nettoyage et validation des données

Points	Critère observable
0	Les données ne sont ni validées ni nettoyées
2	Les valeurs null sont supprimées pour les champs nécessaires et éviter des erreurs (NPE) ou des traitements incohérents
3	Les champs nécessaires sont validés et convertis dans le bon format
5	Les données nettoyées et validées sont stockées dans un output intermédiaire avec un format qui prend en charge le schéma

Orchestration

Points	Critère observable
0	Le job ne peut pas se lancer manuellement via une commande (pas de package exécutable)
2	Le job se lance manuellement via une commande
3	Le pipeline contient plusieurs étapes (jobs) qui s'enchaînent ; à lancer manuellement via une ou plusieurs commandes
5	Le pipeline contient plusieurs étapes et l'enchaînement est automatisé via orchestrateur ou CRON

Source de données externe

Points	Critère observable
0	Le data lake n'est pas enrichi par une source de données externe
2	Le data lake est alimenté par la réponse d'un appel à une URL manuellement
3	Le data lake est alimenté par la réponse traitée d'un appel à une URL manuellement
5	Le data lake est alimenté en continu (avec une architecture utilisant orchestrateur type Airflow ou CRON, websocket, message queue ou autre)

Calcul distribué

Points	Critère observable
0	L'application n'utilise pas de framework de traitement distribué
2	L'application utilise aussi des opérations de filtering ou d'ordering
3	L'application utilise aussi des opérations de grouping ou de window functions
5	L'application utilise aussi des optimisations de la distribution et de répartition de la charge

Utilisation de SQL

Points	Critère observable
0	Le projet n'utilise pas de SQL ou NoSQL
2	L'application ou la requête utilise aussi une requête SQL ou NoSQL pour obtenir un résultat

Points	Critère observable
3	L'application ou la requête utilise aussi une built-in function ou une user-defined function ou une expression en SQL ou NoSQL
5	L'application ou la requête utilise aussi le DDL pour définir un schéma

Les technologies:

- Hadoop et HDFS
- Spark
- Python
- Java
- Crontab