



Projet : Simulation de Production Usine Connectée

Objectif :

Créer une application Python simulant une usine connectée, avec gestion des matériaux, des machines, des flux, des débits de production et des anomalies.

Tâches principales :

1. Initialisation de l'usine

- Créer des classes Python pour représenter :
 - **Usine** : contient les informations générales (nom, localisation, capacité de stockage).
 - **Machine** : a des attributs tels que type, débit maximal, temps de cycle, et état (actif, en panne, en maintenance).
 - **Matériaux** : inclut nom, quantité disponible, et coût.

2. Gestion des recettes et production

- Définir des recettes sous forme de dictionnaires (exemple : Aluminium -> Fourchette nécessite 5 unités d'Aluminium).
- Implémenter une fonction pour :
 - Vérifier la disponibilité des matériaux.
 - Déclencher le processus de production.
 - Simuler le temps requis en fonction des machines disponibles.

3. Gestion des flux et des machines

- Ajouter un mécanisme de file d'attente pour les matières premières à transformer.
- Simuler le fonctionnement de plusieurs machines, en prenant en compte :
 - Le nombre de machines nécessaires pour une recette.
 - Le temps de traitement.
- Mettre en pause la production si une machine est en panne ou en maintenance.

4. Détection et gestion des anomalies

- Créer une fonction pour détecter des anomalies courantes, telles que :
 - **Pannes de machines** : simulation aléatoire.
 - **Manque de matériaux** : message d'alerte.
 - **Dépassement de capacité** : déclenchement d'un état critique.
- Intégrer un journal des anomalies (log des incidents).

5. Visualisation et rapports

- Générer des rapports simples sur :
 - Le taux de production.
 - Les anomalies détectées.
 - L'état des stocks.
- Afficher un tableau de bord en terminal ou avec une bibliothèque graphique comme matplotlib pour visualiser :
 - L'évolution des stocks.
 - Le fonctionnement des machines.



Bonus (si le temps le permet) :

- Ajouter une interface utilisateur simple en terminal avec tkinter ou django.
 - Implémenter une planification des tâches pour optimiser l'utilisation des machines.
-

Livrables attendus :

1. Un script Python fonctionnel avec :
 - Classes bien structurées.
 - Gestion des flux de production.
 - Détection des anomalies.
 - Code commenté
2. Un rapport final montrant les performances de l'usine (temps de traitement, taux de production, etc.).

-

