



**ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR**

Membre de

HONORIS UNITED UNIVERSITIES

École Marocaine des Sciences de l'Ingénieur (EMSI)

Matière : ERP (Enterprise Resource Planning)

Compte rendu de projet

Module de Gestion du matériel

Implémentation d'un module ERP sous Odoo

Réalisé par :

Adil FAHMI

Année universitaire 2025–2026

Table des matières

1	Introduction	2
2	Contexte et problématique	2
3	Objectifs du projet	2
4	Environnement technique	3
5	Architecture générale du projet	3
6	Structure du module	3
7	Modélisation des données	3
8	Interfaces utilisateur	4
9	Illustrations et captures du projet	4
9.1	Initialisation du module	4
9.2	Définition du modèle métier	5
9.3	Gestion des droits d'accès	6
9.4	Vues et menus	7
9.5	Configuration Odoo	8
9.6	Déploiement Docker	9
9.7	Vue formulaire du matériel	10
9.8	Vue liste du matériel	10
9.9	Exécution des conteneurs Docker	11
10	Résultats obtenus	11
11	Limites du projet	12
12	Perspectives d'amélioration	12
13	Conclusion	12

1 Introduction

La gestion du matériel informatique constitue un pilier essentiel du bon fonctionnement des organisations modernes, qu'il s'agisse d'entreprises privées, d'établissements publics ou d'institutions éducatives. Une mauvaise gestion du parc matériel peut engendrer des pertes financières importantes, des interruptions de service, ainsi qu'une baisse de la productivité.

Dans ce contexte, le présent projet vise à concevoir et développer un module personnalisé sous **Odoo 17** permettant la gestion efficace du matériel informatique. Odoo est un progiciel de gestion intégré (ERP) open source, reconnu pour sa modularité et sa flexibilité.

Ce projet s'inscrit dans une démarche pédagogique visant à mettre en pratique les concepts fondamentaux du développement ERP, notamment la modélisation des données, la création d'interfaces utilisateur et le déploiement d'une application dans un environnement conteneurisé.

2 Contexte et problématique

Dans de nombreuses structures, la gestion du matériel informatique est souvent réalisée à l'aide de solutions peu adaptées telles que des fichiers Excel, des documents papier ou des outils non centralisés. Ces méthodes présentent plusieurs limites majeures :

- absence de centralisation et de cohérence des données ;
- difficulté à suivre l'état réel du matériel ;
- manque de traçabilité et d'historique ;
- risques élevés d'erreurs humaines et de pertes d'informations.

Face à ces contraintes, l'intégration de la gestion du matériel dans un ERP tel qu'Odoo apparaît comme une solution pertinente. Elle permet de centraliser l'information, d'améliorer la visibilité sur le parc matériel et de faciliter la prise de décision.

3 Objectifs du projet

L'objectif général de ce projet est de développer un module Odoo fonctionnel et évolutif pour la gestion du matériel informatique.

Les objectifs spécifiques sont les suivants :

- analyser les besoins liés à la gestion du matériel ;
- concevoir un modèle de données adapté ;
- implémenter la logique métier en Python ;
- développer des interfaces utilisateur ergonomiques ;
- déployer l'application dans un environnement Dockerisé ;
- acquérir une expérience concrète du développement sous Odoo.

4 Environnement technique

Le projet repose sur un ensemble de technologies modernes et complémentaires :

- **Odoo 17** comme plateforme ERP ;
- **Python** pour la logique métier ;
- **XML** pour la définition des vues et des menus ;
- **PostgreSQL 16** comme système de gestion de base de données ;
- **Docker et Docker Compose** pour le déploiement.

L'utilisation de Docker permet d'isoler l'environnement d'exécution, de simplifier l'installation du projet et d'assurer une reproductibilité totale sur différentes machines.

5 Architecture générale du projet

L'architecture du projet respecte les standards recommandés par Odoo. Elle repose sur une séparation claire entre :

- la logique métier (modèles Python) ;
- l'interface utilisateur (vues XML) ;
- la configuration et le déploiement (Docker).

Le module est intégré en tant qu'addon personnalisé via le répertoire `extra-addons`, ce qui permet de préserver l'intégrité du noyau Odoo.

6 Structure du module

Le module `tp_gestion_materiel` est organisé selon la structure standard des modules Odoo :

- un dossier `models` contenant la logique métier ;
- un dossier `views` pour les interfaces utilisateur ;
- un dossier `security` pour la gestion des accès ;
- des fichiers de configuration pour le déploiement.

Cette organisation facilite la maintenance et l'évolution future du module.

7 Modélisation des données

Le cœur du projet repose sur le modèle `tp.materiel`, représentant un équipement informatique. Chaque enregistrement correspond à un matériel unique.

Les champs définis permettent de stocker des informations essentielles telles que le nom, la référence, le type, la date d'achat, le coût et l'état du matériel. L'utilisation de champs de type *Selection* garantit la cohérence et la standardisation des données.

8 Interfaces utilisateur

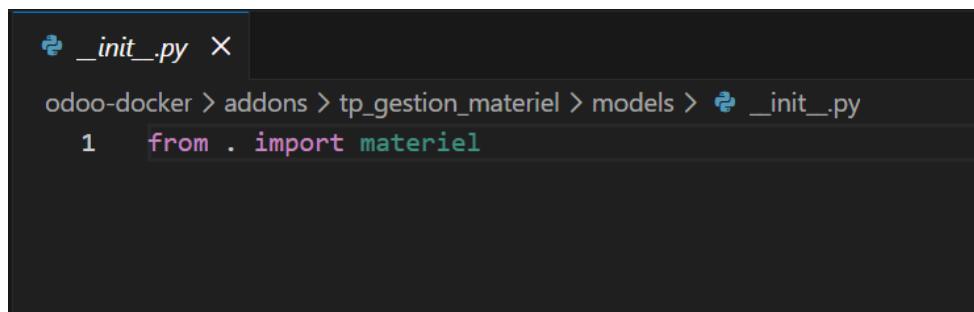
Deux interfaces principales ont été développées afin de répondre aux besoins des utilisateurs :

- **Vue liste (Tree View)** : permet une consultation rapide et globale du parc matériel ;
- **Vue formulaire (Form View)** : permet l'ajout, la modification et la consultation détaillée des équipements.

L'ergonomie des vues a été pensée afin de faciliter la prise en main par des utilisateurs non techniques.

9 Illustrations et captures du projet

9.1 Initialisation du module



```
__init__.py ×
odoo-docker > addons > tp_gestion_materiel > models > __init__.py
1   from . import materiel
```

FIGURE 1 – Fichier `__init__.py` du module

Ce fichier assure l'initialisation du module et l'importation des modèles Python nécessaires à son fonctionnement. Il permet à Odoo de reconnaître correctement le module personnalisé et de charger la logique métier définie dans les fichiers du dossier `models`. Cette étape est indispensable pour l'intégration du module dans l'ERP et garantit son bon chargement lors du démarrage de l'application.

9.2 Définition du modèle métier

```
materiel.py 1 X
odoo-docker > addons > tp_gestion_materiel > models > materiel.py > TpMateriel
1   from odoo import models, fields
2
3
4   class TpMateriel(models.Model):
5       _name = "tp.materiel"
6       _description = "Gestion du Matériel"
7
8       name = fields.Char(string="Nom du matériel", required=True)
9       reference = fields.Char(string="Référence Série")
10
11      type_materiel = fields.Selection([
12          ('pc', 'Ordinateur'),
13          ('ecran', 'Écran'),
14          ('imprimante', 'Imprimante'),
15          ('accessoire', 'Accessoire')
16      ], string="Type", default='pc')
17
18      date_achat = fields.Date(string="Date d'achat")
19      cout = fields.Float(string="Coût d'achat")
20
21      etat = fields.Selection([
22          ('neuf', 'Neuf'),
23          ('bon', 'Bon état'),
24          ('panne', 'En panne'),
25          ('rebut', 'Au rebut')
26      ], string="État", default='neuf')
27
28      description = fields.Text(string="Notes")
29
```

FIGURE 2 – Définition du modèle métier `tp.materiel`

Cette capture illustre la définition du modèle métier principal du projet. Le modèle `tp.materiel` est implémenté en Python et hérite de la classe `models.Model`, conformément aux standards de développement Odoo. Il regroupe l'ensemble des champs nécessaires à la description d'un équipement informatique, tels que le nom, le type, la référence, la date d'achat, le coût et l'état. Cette modélisation constitue le cœur fonctionnel du module et permet une gestion structurée et cohérente des données.

9.3 Gestion des droits d'accès

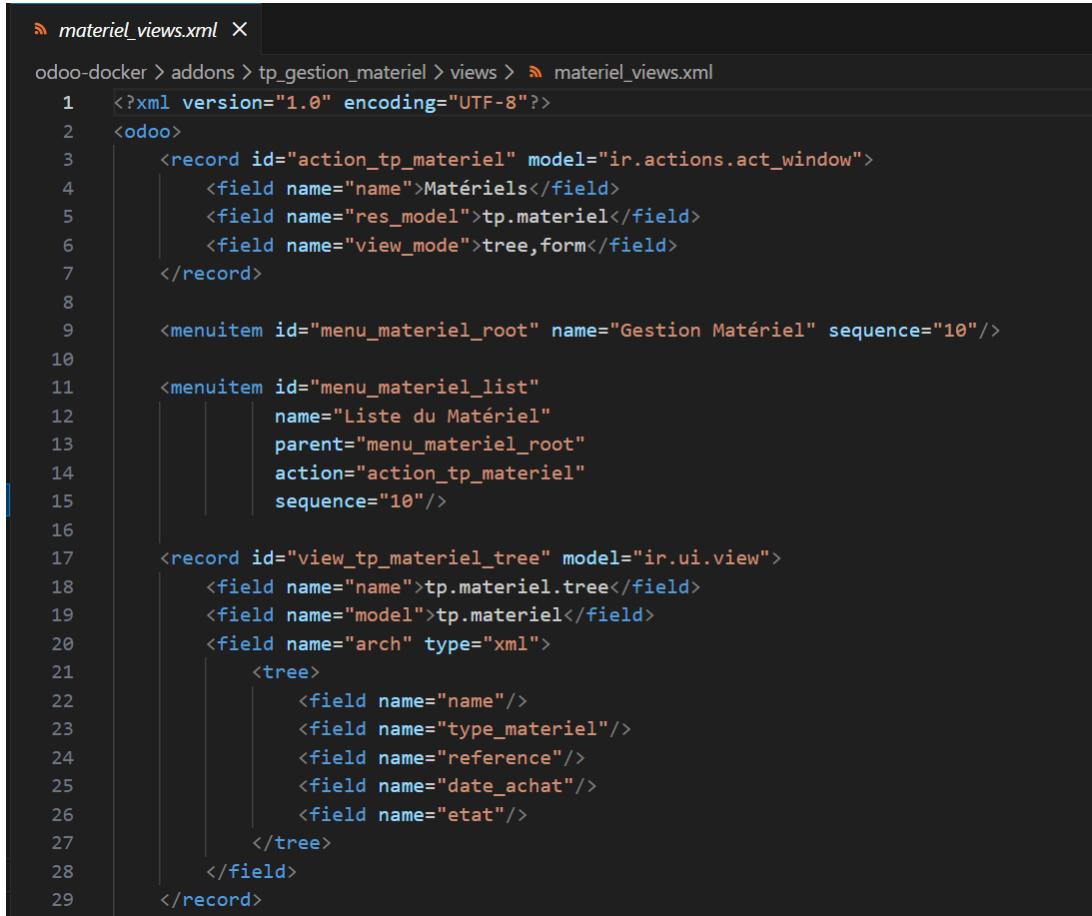


```
ir.model.access.csv X
odoo-docker > addons > tp_gestion_materiel > security > ir.model.access.csv > data
1   id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
2   access_tp_materiel,tp.materiel,model_tp_materiel,,1,1,1,1
```

FIGURE 3 – Fichier de sécurité ir.model.access.csv

Ce fichier définit les règles de sécurité appliquées au modèle `tp.materiel`. Il précise les permissions accordées aux utilisateurs concernant la lecture, l'écriture, la création et la suppression des enregistrements. La mise en place de ce fichier permet de contrôler l'accès aux données et d'assurer un niveau minimal de sécurité au sein de l'application. Il s'agit d'un élément essentiel pour toute application Odoo destinée à un environnement multi-utilisateurs.

9.4 Vues et menus

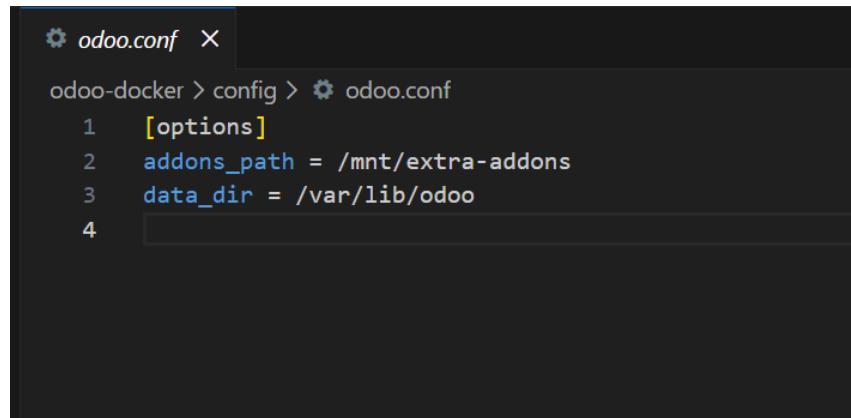


```
materiel_views.xml ×
odoo-docker > addons > tp_gestion_materiel > views > materiel_views.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <odoor>
3      <record id="action_tp_materiel" model="ir.actions.act_window">
4          <field name="name">Matériels</field>
5          <field name="res_model">tp.materiel</field>
6          <field name="view_mode">tree,form</field>
7      </record>
8
9      <menuitem id="menu_materiel_root" name="Gestion Matériel" sequence="10"/>
10
11     <menuitem id="menu_materiel_list"
12         name="Liste du Matériel"
13         parent="menu_materiel_root"
14         action="action_tp_materiel"
15         sequence="10"/>
16
17     <record id="view_tp_materiel_tree" model="ir.ui.view">
18         <field name="name">tp.materiel.tree</field>
19         <field name="model">tp.materiel</field>
20         <field name="arch" type="xml">
21             <tree>
22                 <field name="name"/>
23                 <field name="type_materiel"/>
24                 <field name="reference"/>
25                 <field name="date_achat"/>
26                 <field name="etat"/>
27             </tree>
28         </field>
29     </record>
```

FIGURE 4 – Définition des vues XML et des menus

Cette capture présente la définition des vues XML ainsi que la configuration des menus du module. Elle permet de créer l'action associée au modèle, le menu principal « Gestion Matériel » ainsi que les vues nécessaires à l'affichage des données. Grâce à ces éléments, les utilisateurs peuvent accéder facilement au module depuis l'interface Odoo et interagir avec les données de manière intuitive. Les vues XML assurent la liaison entre la logique métier et l'interface graphique.

9.5 Configuration Odoo

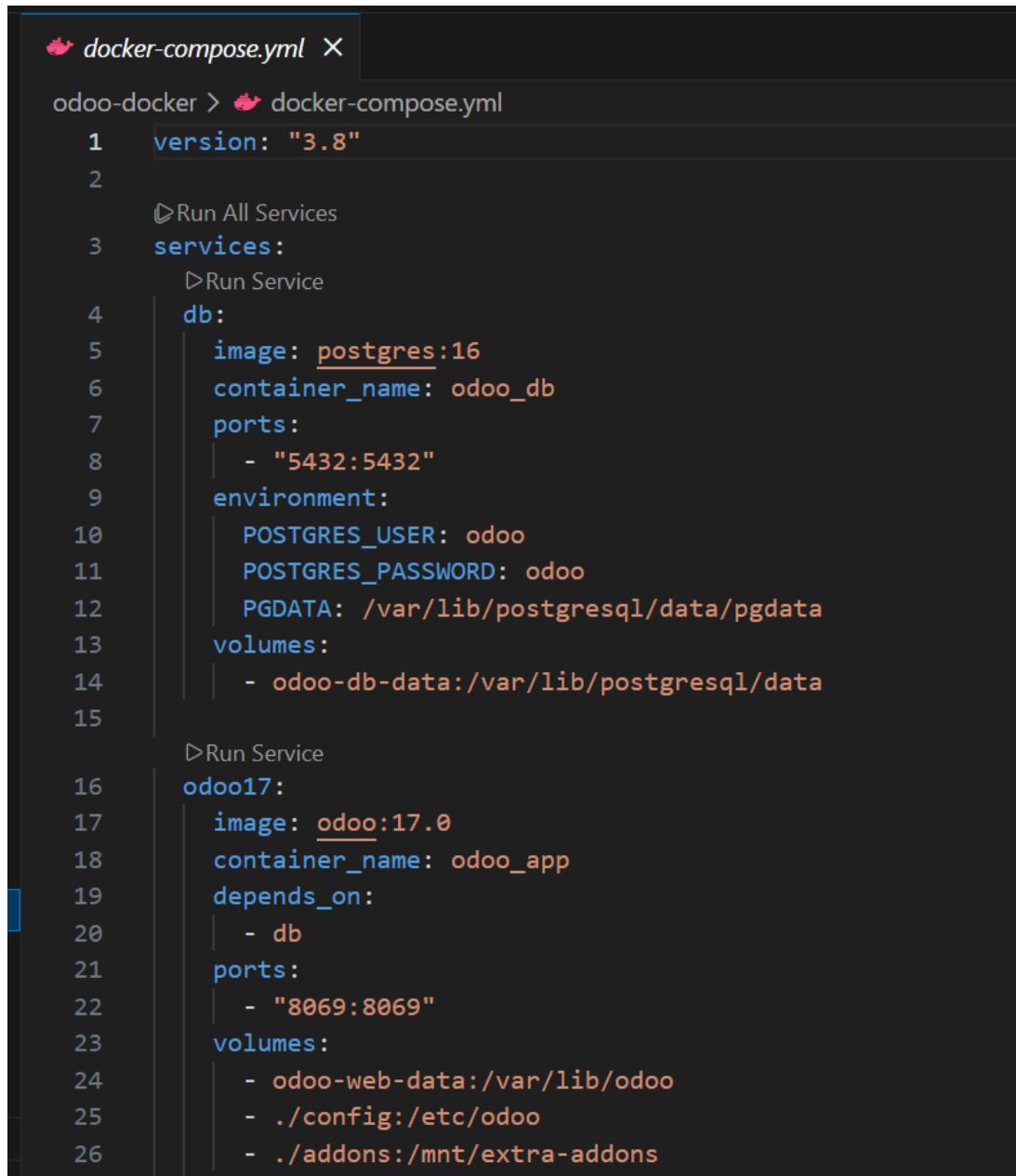


```
odoo.conf X
odoo-docker > config > odoo.conf
1 [options]
2 addons_path = /mnt/extra-addons
3 data_dir = /var/lib/odoo
4
```

FIGURE 5 – Fichier de configuration odoo.conf

Ce fichier contient les paramètres essentiels de configuration d’Odoo. Il définit notamment le chemin des addons personnalisés ainsi que le répertoire de stockage des données. Cette configuration permet à Odoo de charger correctement le module développé et d’assurer le bon fonctionnement de l’application dans l’environnement Dockerisé. Elle joue un rôle clé dans l’intégration du module au sein de l’ERP.

9.6 Déploiement Docker



```
docker-compose.yml X
odoo-docker > docker-compose.yml
1 version: "3.8"
2
3   ▷Run All Services
4   services:
5     db:
6       image: postgres:16
7       container_name: odoo_db
8       ports:
9         - "5432:5432"
10      environment:
11        POSTGRES_USER: odoo
12        POSTGRES_PASSWORD: odoo
13        PGDATA: /var/lib/postgresql/data/pgdata
14      volumes:
15        - odoo-db-data:/var/lib/postgresql/data
16
17      ▷Run Service
18      odoo17:
19        image: odoo:17.0
20        container_name: odoo_app
21        depends_on:
22          - db
23        ports:
24          - "8069:8069"
25        volumes:
26          - odoo-web-data:/var/lib/odoo
27          - ./config:/etc/odoo
28          - ./addons:/mnt/extras-addons
```

FIGURE 6 – Fichier docker-compose.yml

Cette capture illustre le fichier `docker-compose.yml` utilisé pour le déploiement du projet. Il définit les différents services nécessaires au fonctionnement de l'application, notamment Odoo 17 et la base de données PostgreSQL. L'utilisation de Docker Compose permet de lancer l'ensemble de l'environnement avec une seule commande, tout en garantissant l'isolation, la portabilité et la reproductibilité du système. Cette approche facilite le déploiement et la maintenance de l'application.

9.7 Vue formulaire du matériel

The screenshot shows a detailed form for a computer. At the top, there are buttons for 'New' and 'Matériels'. The main section is titled 'Pc HP' and contains the following data:

Type	Ordinateur	Date d'achat	01/01/2026
Référence Série		Coût d'achat	100.00
État	Neuf		

Below this, there is a 'Description' field containing 'En cours' and a scrollable area.

FIGURE 7 – Vue formulaire d'un matériel informatique dans Odoo

Cette vue formulaire permet de consulter et de modifier l'ensemble des informations relatives à un matériel informatique. Elle offre une présentation détaillée des données, avec une organisation claire des champs facilitant la saisie et la lecture des informations. Cette interface constitue l'élément central pour la gestion opérationnelle du matériel, car elle permet aux utilisateurs d'effectuer toutes les actions de gestion sur un équipement donné. Son ergonomie contribue à une meilleure expérience utilisateur.

9.8 Vue liste du matériel

The screenshot shows a list view of materials. At the top, there are buttons for 'New' and 'Matériels'. A search bar is present, and the status is '1-1 / 1'. The table has columns: Nom du matériel, Type, Référence Série, Date d'achat, and État. One row is visible:

Nom du matériel	Type	Référence Série	Date d'achat	État
Pc HP	Ordinateur		01/01/2026	Neuf

FIGURE 8 – Vue liste (Tree View) du matériel informatique

La vue liste présente l'ensemble du parc matériel sous forme tabulaire. Elle permet une vision globale et synthétique des équipements enregistrés dans le système, en affichant les informations essentielles telles que le nom du matériel, le type, la date d'achat et l'état. Cette vue facilite le suivi quotidien du matériel, la recherche rapide d'un équipement et l'accès direct aux fiches détaillées. Elle constitue un outil indispensable pour une gestion efficace du parc informatique.

9.9 Exécution des conteneurs Docker

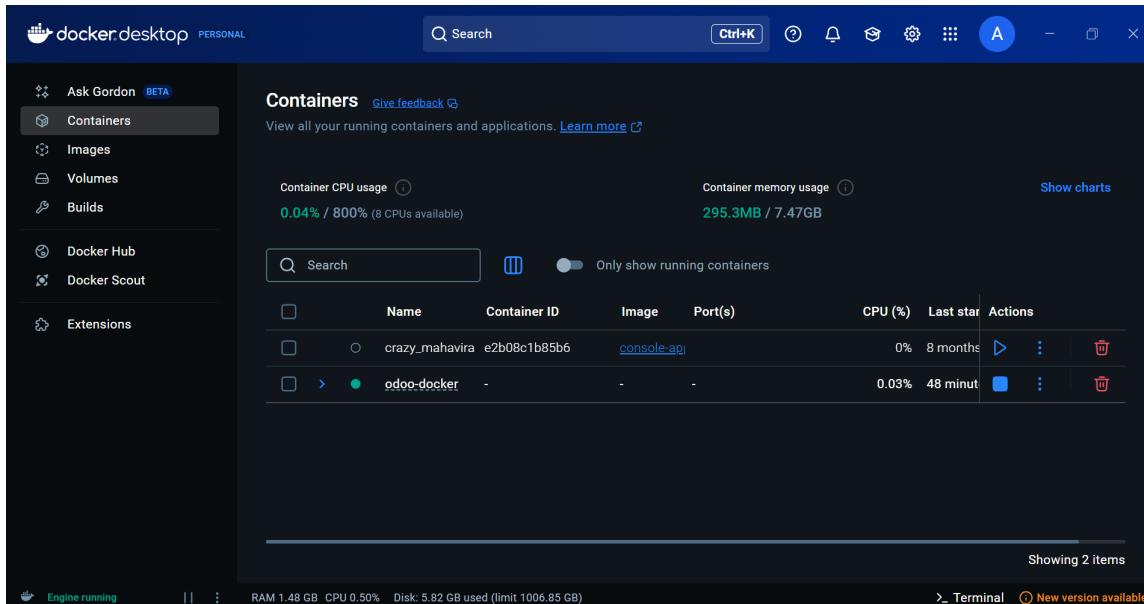


FIGURE 9 – Exécution des conteneurs du projet via Docker Desktop

Cette capture d'écran présente l'interface Docker Desktop affichant les conteneurs en cours d'exécution pour le projet. Elle permet de vérifier l'état des services Odoo et PostgreSQL, confirmant que l'environnement est correctement déployé et opérationnel. Cette visualisation constitue une preuve du bon fonctionnement de l'architecture Docker mise en place et valide l'exécution réelle de l'application dans un environnement conteneurisé.

10 Résultats obtenus

À l'issue du projet, le module développé est pleinement fonctionnel et répond aux objectifs définis au début de l'étude. Il permet une gestion complète et structurée du matériel informatique, incluant la création, la consultation, la modification et le suivi des équipements au sein de l'application Odoo.

Les données saisies sont stockées de manière persistante dans la base de données PostgreSQL, garantissant leur fiabilité et leur conservation même après le redémarrage des services. L'interface utilisateur proposée, basée sur les vues liste et formulaire d'Odoo, est claire, ergonomique et facile à prendre en main, ce qui facilite l'utilisation du module par des utilisateurs non techniques.

Par ailleurs, le déploiement du projet dans un environnement Dockerisé assure une exécution stable et reproductible de l'application. Les tests réalisés confirment le bon fonctionnement des différentes fonctionnalités développées ainsi que l'intégration correcte du module au sein de l'ERP Odoo.

11 Limites du projet

Certaines limites ont été identifiées :

- absence de gestion avancée des rôles utilisateurs ;
- absence de reporting et de tableaux de bord ;
- fonctionnalités limitées au périmètre de base.

12 Perspectives d'amélioration

Plusieurs pistes d'évolution peuvent être envisagées :

- attribution du matériel aux utilisateurs ou services ;
- ajout de rapports PDF automatisés ;
- intégration d'une vue Kanban ;
- mise en place d'un historique des modifications.

13 Conclusion

Ce projet a permis de concevoir et de développer un module Odoo complet dédié à la gestion du matériel informatique, en respectant les bonnes pratiques de développement propres à l'ERP Odoo. L'ensemble des objectifs fixés au début du projet a été atteint, notamment la modélisation des données, la création d'interfaces utilisateur ergonomiques et le déploiement de l'application dans un environnement Dockerisé.

Au-delà de l'aspect fonctionnel, ce projet a constitué une expérience très enrichissante sur le plan technique et méthodologique. Il m'a permis de consolider mes compétences en développement Python, en conception de modules ERP, en structuration de bases de données et en utilisation des technologies de conteneurisation telles que Docker et Docker Compose. La mise en œuvre de ce module m'a également permis de mieux comprendre le fonctionnement interne d'Odoo, ainsi que les interactions entre la logique métier, l'interface utilisateur et l'environnement de déploiement.

Ce travail constitue une base solide pour des développements futurs plus avancés dans un contexte professionnel. Les perspectives d'amélioration identifiées, telles que l'ajout de fonctionnalités de reporting, la gestion avancée des utilisateurs ou l'intégration de tableaux de bord, ouvrent la voie à l'évolution du module vers une solution plus complète et adaptée aux besoins réels des organisations.

Enfin, ce projet représente une première expérience concrète et significative dans le développement de solutions ERP, renforçant ma préparation à intégrer des projets professionnels nécessitant rigueur, analyse et maîtrise des outils technologiques modernes.