

Rapport d'Iteration – Projet Application de Gestion de Tâches

Projet : Application de Gestion de Tâches

Itération : 2

Groupe : Marchal Enzo, Nacime Laghezali, Cabot Matthieu, Vincent Julien

Date : 15/12/2025

1. Objectif de l'itération

L'objectif principal de cette seconde itération était de consolider l'architecture logicielle et d'introduire la persistance des données, tout en initiant la transition vers une interface graphique.

Plus précisément, cette itération visait à :

- **Structurer l'application selon le patron MVC (Modèle-Vue-Contrôleur)** pour séparer clairement les responsabilités.
- **Mettre en place la persistance des données (DAO)** afin que les projets et tâches soient sauvegardés en base de données relationnelle.
- **Développer la première ébauche de l'interface graphique (JavaFX)** pour visualiser les projets sous forme de cartes (Kanban).
- **Finaliser l'interface textuelle** pour valider l'ensemble des fonctionnalités via le contrôleur.

2. Fonctionnalités et cas d'utilisation

2.1 Liste des fonctionnalités développées

Mise en place du DAO (Data Access Object)

- **Objectif** : Assurer la persistance des données (Projets, Colonnes, Tâches) dans une base de données MySQL, remplaçant le stockage volatile en mémoire.
- **Développement** : Julien Vincent (JV)

Architecture MVC (Modèle-Vue-Contrôleur)

- **Objectif** : Réorganiser le code avec le patron d'architecture MVC pour faciliter l'ajout de nouvelle vue.
- **Développement** : Nacime Laghezali (NL)

Vue Texte (Finalisation et Liaison MVC)

- **Objectif** : Permettre l'interaction complète avec le système via la console en utilisant la nouvelle architecture MVC.
- **Développement** : Enzo Marchal (EM)

Suppression de tâche (Vue kanban)

- **Objectif** : Supprimer une tâche spécifique sur la vue kanban
- **Développement** : Matthieu Cabot (MC)

Vue Kanban JFX (Initiation)

- **Objectif** : Créer la fenêtre principale de l'application graphique ("Dashboard").
- **Développement** : Matthieu Cabot (MC)

3. Diagramme de classes et choix de conception

3.1 Nouveaux Choix de Conception

Patron MVC (Modèle-Vue-Contrôleur)

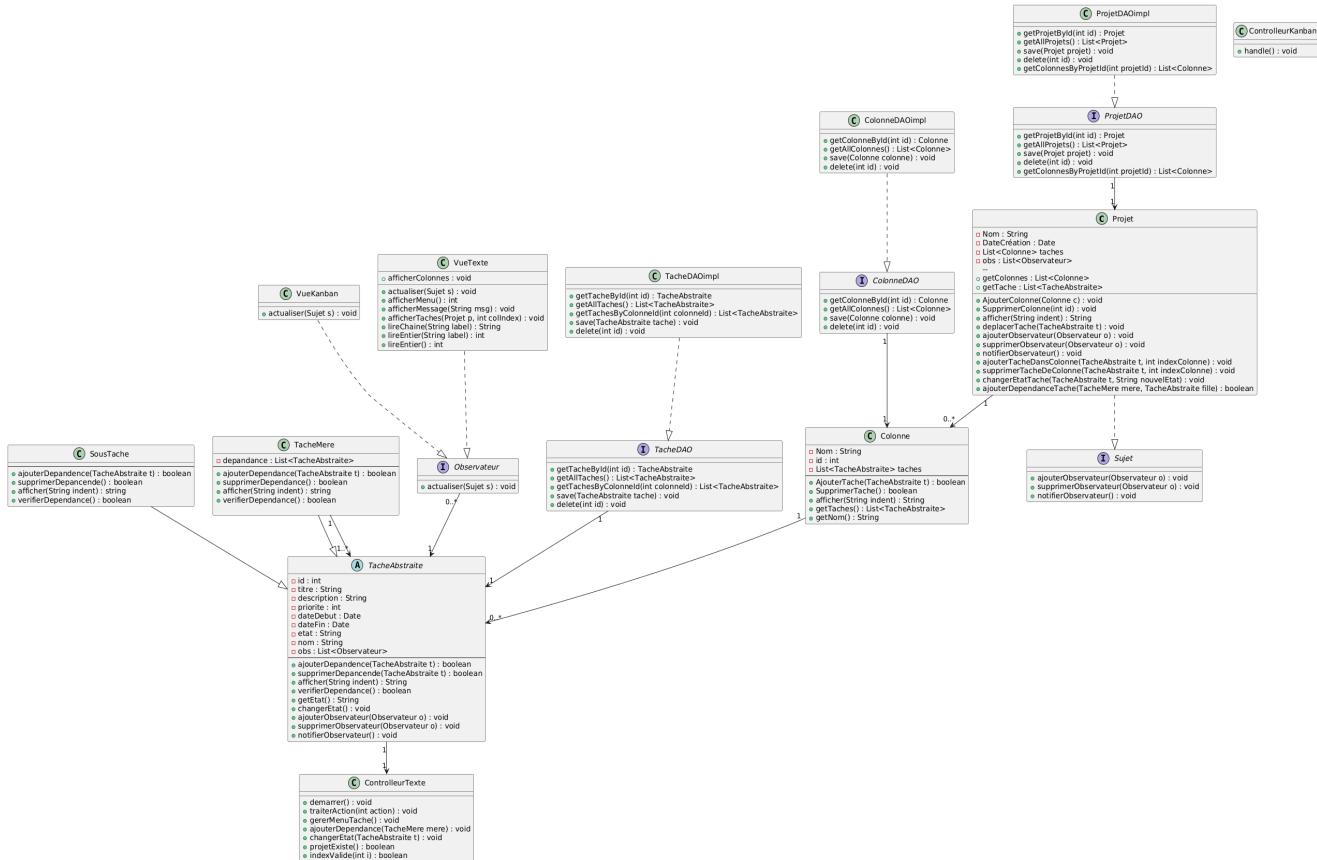
Nous avons séparé le projet en trois couches distinctes :

- **Modèle** : La classe Projet.
- **Vue** : Les interfaces (VueTexte et la nouvelle VueDashboard JavaFX).
- **Contrôleur** : Le chef d'orchestre (ControleurTexte) qui reçoit les entrées utilisateur et modifie le modèle.

Patron DAO (Data Access Object)

Pour isoler la logique métier des détails de la base de données (SQL), nous avons implémenté le pattern DAO :

- Chaque entité métier possède son interface (IProjetDAO, ITacheDAO) et son implémentation (ProjetDAOImpl, TacheDAOImpl).
- La connexion est gérée par un Singleton (DBConnection) pour éviter les connexions multiples inutiles.



4. Bilan de l'itération

Cette deuxième itération a été cruciale pour la viabilité technique du projet. Nous sommes passés d'un modèle objet en mémoire à une application persistante structurée.

Réalisations majeures :

- Persistance réussie** : L'application sauvegarde et recharge désormais l'état des projets grâce à l'implémentation des DAO par Julien.
- Architecture MVC** : La mise en place du MVC par Nacime facilite grandement l'ajout de la nouvelle vue graphique sans casser la vue textuelle existante.
- Début de l'UI** : La vue Kanban initiée par Matthieu pose les bases visuelles de l'application finale.