

Cahier des Charges

Développement d'une API de Gestion des Tâches

1 Contexte et Objectif

Ce projet vise à développer une API REST pour la gestion des tâches au sein d'une équipe. L'objectif est de permettre aux managers de créer et assigner des tâches, et aux employés de gérer leur travail avec un système de suivi d'état et de validation.

2 Description Fonctionnelle

L'application comporte 2 rôles : **Manager** et **Employé**.

2.1 Manager

- **Créer une tâche** : titre, description, date limite, priorité (Basse/Moyenne/Haute/Urgente), employé assigné, code unique auto-généré
- **Consulter et filtrer** toutes les tâches par employé, état, priorité et date limite
- **Modifier** une tâche si état = « À faire » ou « En cours »
- **Supprimer** une tâche si état = « À faire »
- **Valider/Rejeter** les tâches terminées (rejet = retour en cours + commentaire)
- **Consulter l'historique** des modifications

2.2 Employé

- **Consulter** ses tâches assignées avec filtres
- **Mettre à jour l'état** : À faire → En cours → Terminée, ou marquer comme Bloquée (commentaire obligatoire)
- **Ajouter des commentaires** pour informer le manager
- **Consulter les détails** : description, deadline, priorité, historique

2.3 Règles de Gestion

- Une tâche = un seul employé | Un employé = plusieurs tâches
- États possibles : **À faire**, **En cours**, **Terminée**, **Bloquée**, **Validée**
- Une tâche validée est clôturée et non modifiable
- Chaque tâche affiche l'employé assigné et le manager créateur

3 Architecture Technique

3.1 Technologies

- | | |
|----------------------------|---------------------------|
| — ASP.NET Core 8.0 Web API | — SQL Server |
| — C# | — Swagger |
| — Entity Framework Core | — Architecture en couches |

3.2 Structure de l'API

Authentification : Login et gestion des sessions

Endpoints Manager :

- CRUD des tâches, Consultation/filtrage, Validation/rejet

Endpoints Employé :

- Consultation des tâches assignées, Mise à jour du statut, Ajout de commentaires