



5<sup>ème</sup> année Ingénierie Informatique et Réseaux (5IIR)

## Rapport Projet Odoo

Développement d'un module Odoo  
pour la gestion participative des idées

Réalisé par :

Ilyas Ankri

Sous l'encadrement de :

Mr.Mohammed AitDaoud

Année universitaire :

2025 - 2026



# Remerciements

Au terme de ce travail, nous tenons à exprimer notre profonde gratitude et nos sincères remerciements à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce projet.

Nous adressons tout d'abord nos remerciements les plus chaleureux à notre professeur et encadrant, Monsieur Mohammed Daoud. Nous lui sommes reconnaissants pour la qualité exceptionnelle de son enseignement et pour son expertise dans le domaine des systèmes ERP. Sa pédagogie, sa disponibilité et ses conseils avisés nous ont non seulement permis de mener à bien le développement de ce module Odoo, mais aussi d'acquérir une vision professionnelle indispensable pour notre future carrière d'ingénieurs.

Nos vifs remerciements vont également à notre école, l'EMSI (École Marocaine des Sciences de l'Ingénieur). Nous sommes fiers d'appartenir à cette institution prestigieuse qui nous offre un cadre d'apprentissage stimulant et une formation d'excellence. Merci au corps professoral et administratif de l'EMSI pour les moyens mis à notre disposition et pour leur engagement constant à former des ingénieurs compétents et innovants.

Enfin, nous remercions nos familles et nos proches pour leur soutien inconditionnel et leurs encouragements tout au long de notre parcours académique.

# Dédicace

Au seuil de l'accomplissement que représente ce projet, il nous est essentiel de marquer une pause pour rendre hommage à ceux sans qui cette aventure humaine et intellectuelle n'aurait jamais pu voir le jour. Ce travail est bien plus que la somme de nos efforts techniques sur Odoo ; il est la convergence de l'amour, du soutien et des sacrifices de nos proches.

À nos très chers parents, Nos premiers modèles et nos piliers indéfectibles. Nous peinons à trouver les mots justes pour exprimer l'immensité de notre gratitude. Depuis nos premiers pas, vous avez été les artisans de notre éducation, nous inculquant les valeurs de persévérance et de travail qui nous animent aujourd'hui. Vous avez consenti à d'innombrables sacrifices pour paver la voie de notre avenir d'ingénieurs. Votre confiance en nos capacités a été notre moteur le plus puissant dans les moments de doute. Puisse ce travail être un modeste reflet de tout ce que vous nous avez donné et le témoignage de notre désir le plus cher : celui de vous rendre fiers.

À nos frères et sœurs, Nos confidents et nos alliés de toujours. Merci pour votre patience infinie durant nos longues heures de code et de rédaction, pour votre complicité qui nous a offert des bouffées d'air frais, et pour cet humour qui a si souvent désamorcé notre stress. Votre soutien a été une force tranquille qui nous a portés tout au long de ce parcours.

À nos amis, Cette famille que l'on se choisit. À ceux qui ont vécu cette aventure à nos côtés au sein de l'EMSI, qui ont supporté nos absences, écouté nos doutes et su trouver les mots pour nous remotiver. Ce projet porte aussi l'empreinte de votre solidarité et de votre amitié précieuse.

À vous tous, Qui avez cru en nous avant même que nous n'osions le faire, nous dédions l'aboutissement de ces années d'efforts.

Ce succès est le vôtre.

# Résumé

Dans un environnement concurrentiel où l'innovation est un moteur de croissance, l'intelligence collective des collaborateurs représente une ressource inestimable. Pourtant, faute d'outils adaptés, de nombreuses idées pertinentes émanant du terrain sont perdues, oubliées ou ne parviennent jamais aux décideurs. Les boîtes à idées physiques ou les simples échanges d'emails s'avèrent inefficaces pour structurer, suivre et valoriser ces initiatives.

Pour répondre à ce besoin, notre projet porte sur la conception et le développement d'un module de "Boîte à Idées" (Idea Box) intégré à l'ERP Odoo. Cette solution vise à digitaliser le processus d'innovation participative en offrant une plateforme centralisée où chaque collaborateur peut proposer des améliorations, classées par catégorie (Tech, RH, Bureau, etc.). Le module intègre un système de vote interactif pour faire émerger les meilleures propositions et un workflow de validation (Brouillon, Confirmé, Réalisé) pour assurer leur suivi opérationnel.

Techniquement, le module s'appuie sur l'architecture MVC d'Odoo. Le backend en Python gère la logique métier (calcul des votes, contraintes), tandis que le frontend exploite les vues XML (Kanban, Graphique, Liste) pour offrir une expérience utilisateur ludique et ergonomique. L'ensemble est déployé via Docker pour garantir sa portabilité. Ce projet permet ainsi de stimuler l'engagement des employés et de transformer les idées brutes en projets concrets pour l'entreprise.

## Mots-clés du projet :

Odoo, Innovation Participative, Boîte à Idées, Python/XML, Workflow, Vote, Kanban.

# Abstract

In a competitive environment where innovation drives growth, the collective intelligence of employees is an invaluable resource. However, due to a lack of suitable tools, many relevant ideas from the field are lost, forgotten, or never reach decision-makers. Physical suggestion boxes or simple email exchanges prove ineffective for structuring, tracking, and valuing these initiatives.

To address this need, our project focuses on the design and development of an "Idea Box" module integrated into the Odoo ERP. This solution aims to digitalize the participatory innovation process by providing a centralized platform where every employee can propose improvements, classified by category (Tech, HR, Office, etc.). The module integrates an interactive voting system to highlight the best proposals and a validation workflow (Draft, Confirmed, Done) to ensure their operational tracking.

Technically, the module relies on Odoo's MVC architecture. The Python backend manages business logic (vote calculation, constraints), while the frontend leverages XML views (Kanban, Graph, List) to offer a playful and ergonomic user experience. The entire solution is deployed via Docker to ensure portability. This project thus stimulates employee engagement and transforms raw ideas into concrete projects for the company.

## **Project Keywords :**

Odoo, Participatory Innovation, Idea Box, Python/XML, Workflow, Voting, Kanban.

# Table des matières

<b>Remerciements</b>	<b>3</b>
<b>Dédicace</b>	<b>4</b>
<b>Résumé</b>	<b>5</b>
<b>Abstract</b>	<b>6</b>
<b>Chapitre 1 : Contexte général du projet</b>	<b>3</b>
1    Présentation du projet . . . . .	3
2    Objectif du projet . . . . .	3
3    Missions et tâches . . . . .	3
4    Conclusion . . . . .	3
<b>Chapitre 2 : Spécification et analyse</b>	<b>6</b>
1    Docker . . . . .	7
2    Visual Studio Code . . . . .	7
3    Odoo . . . . .	8
4    Git & Git Bash . . . . .	9
5    Python . . . . .	11
<b>Chapitre 3 : Conception et modélisation</b>	<b>14</b>
1    Diagramme de cas d'utilisation . . . . .	14
2    Diagramme de séquence . . . . .	14
3    Diagramme de classe . . . . .	16
<b>Chapitre 4 : Réalisation et mise en œuvre</b>	<b>20</b>
1    Vue Kanban - Gestion visuelle du flux d'idées . . . . .	20
2    Vue Liste - Supervision globale et classement . . . . .	20
3    Vue Graphique - Analyse statistique des tendances . . . . .	21
4    Vue Formulaire - Détail et cycle de vie de l'idée . . . . .	21

# Table des figures

1	Logo Docker . . . . .	7
2	Logo Visual Studio Code . . . . .	8
3	Logo Odoo . . . . .	9
4	Logo Git & Git Bash . . . . .	10
5	Logo Python . . . . .	11
6	diagramme de cas d'utilisation . . . . .	15
7	Diagramme de séquence . . . . .	16
8	Diagramme de classe . . . . .	17
9	Vue Kanban - Gestion visuelle du flux d'idées . . . . .	20
10	Vue Liste - Supervision globale et classement . . . . .	21
11	Vue Graphique - Analyse statistique des tendances . . . . .	21
12	Vue Formulaire - Détail et cycle de vie de l'idée . . . . .	22



# Introduction générale

L'innovation n'est plus l'apanage des seuls départements RD ; elle est l'affaire de tous. Dans les entreprises modernes, chaque collaborateur, de par sa proximité avec les processus quotidiens ou les clients, est susceptible de formuler des propositions d'amélioration pertinentes. Cependant, capter cette "voix de l'employé" reste un défi majeur. Les processus informels manquent de transparence et de suivi, ce qui engendre frustration et désengagement.

C'est dans ce contexte que s'inscrit la mise en place d'un système de gestion des idées. L'intégration d'un tel outil au sein de l'ERP de l'entreprise permet non seulement de centraliser les données, mais aussi de lier directement les initiatives aux processus métiers existants.

Notre projet de fin d'année consiste à développer un module Odoo nommé « Boîte à Idées ». L'objectif est de créer un espace numérique démocratique où les idées peuvent être soumises, débattues via un système de vote, et traitées par l'administration. Contrairement à une simple liste de tâches, ce module met l'accent sur l'aspect visuel et interactif (Vue Kanban, Graphiques) pour encourager l'adoption par les utilisateurs.

Ce rapport détaille les étapes de réalisation de ce module, depuis l'analyse des besoins fonctionnels (catégorisation, workflow d'approbation) jusqu'à l'implémentation technique utilisant les langages Python et XML, le tout orchestré dans un environnement conteneurisé Docker.

# Chapitre 1 : Contexte général du projet

# 1 Présentation du projet

Notre projet PFA consiste à développer un module intégré à l'ERP Odoo pour la gestion du cycle de vie des idées. Il répond au besoin des managers de structurer l'innovation interne et au besoin des collaborateurs de se sentir écoutés. Le module permet de remplacer les processus disparates par une application unique capable de gérer la soumission d'une idée, son classement par thématique (RH, Technique, Bureau, etc.), son évaluation par les pairs (système de votes) et sa validation par l'administration.

## 2 Objectif du projet

L'objectif principal est de fournir un outil ludique et efficace pour démocratiser l'innovation. Les objectifs spécifiques sont :

Centraliser les initiatives : Stocker toutes les propositions dans une base de données unique PostgreSQL.

Encourager l'interaction : Mettre en place un mécanisme de « Vote » (+1) permettant de faire émerger les idées les plus populaires.

Assurer le suivi visuel : Utiliser une vue Kanban (cartes déplaçables) pour visualiser l'état d'avancement des idées (Brouillon, Confirmé, En cours, Réalisé).

Analyser les tendances : Fournir des vues graphiques pour identifier les catégories qui suscitent le plus d'intérêt.

## 3 Missions et tâches

Dans le cadre de ce projet, les missions techniques attribuées à notre binôme sont les suivantes :

Modélisation des données : Définir l'entité `idea.box` avec ses attributs (Titre, Auteur, Date, Catégorie, Description, Score).

Développement Backend (Python) : Implémenter la logique des compteurs de votes (méthode d'incrémentation) et les transitions du workflow (changement d'état).

Conception des Vues Avancées (XML) :

Création d'une vue Kanban ergonomique avec code couleur.

Création d'une vue Graphique pour les statistiques par catégorie.

Création d'une vue Formulaire claire pour la saisie.

Déploiement Docker : Configuration de l'environnement conteneurisé pour assurer la portabilité du module.

## 4 Conclusion

En guise de conclusion, ce premier chapitre a permis de poser le cadre général de notre projet de fin d'année. Nous avons mis en lumière les limites des méthodes traditionnelles de gestion des emplois du temps et souligné l'importance stratégique de la digitalisation des processus académiques.

Face à ces enjeux, notre solution, matérialisée par le développement d'un module intégré à l'ERP Odoo, apparaît comme une réponse pertinente et robuste. Nous avons défini les objectifs principaux du projet, visant à centraliser l'information, sécuriser les données et automatiser la planification. De plus, l'identification claire des missions et des choix technologiques (Python, XML, PostgreSQL, Docker) nous offre une feuille de route précise pour les phases de réalisation.

Ce cadrage contextuel et fonctionnel étant désormais établi, le chapitre suivant sera consacré à l'étude détaillée des outils de développement et à l'analyse technique qui structureront la mise en œuvre de notre solution.

## Chapitre 2 : Spécification et analyse

## Introduction

La phase de conception permet de traduire le besoin de "gestion collaborative" en une structure technique cohérente. Pour ce module "Boîte à Idées", l'accent a été mis sur la simplicité d'interaction (pour encourager les votes) et la clarté du workflow de validation.

## 1 Docker

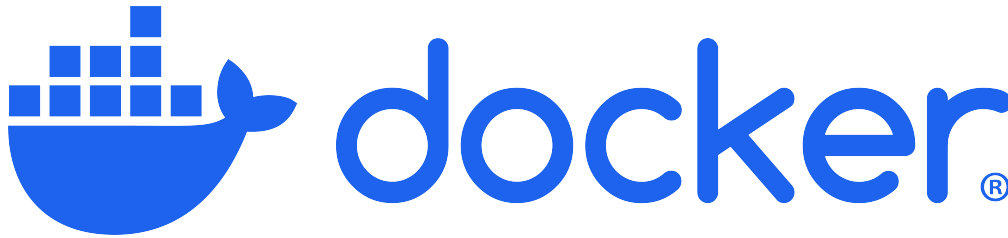


FIGURE 1 – Logo Docker

Docker est une technologie de conteneurisation incontournable dans le développement moderne, facilitant le déploiement d'applications dans des environnements isolés. Contrairement aux machines virtuelles classiques, Docker encapsule l'application Odoo et ses dépendances (bibliothèques Python, système) dans des conteneurs légers, garantissant un comportement identique entre la machine de développement et le serveur de production.

Dans le cadre de ce projet, Docker a été utilisé pour orchestrer l'environnement complet de notre module de gestion de planning. Il nous a permis d'instancier facilement le conteneur du serveur Odoo (version 17) et celui de la base de données PostgreSQL, assurant leur communication via un réseau virtuel dédié. Cette approche a grandement simplifié l'installation, évitant les conflits de versions de Python ou de bibliothèques sur la machine hôte. +1

Grâce à Docker, nous avons pu définir une infrastructure reproductible via le fichier `docker-compose.yml`, permettant de redémarrer les services rapidement après chaque modification majeure du code Python ou des fichiers de configuration. Les conteneurs assurent également une isolation parfaite des données académiques stockées dans les volumes Docker, garantissant leur persistance même en cas d'arrêt des services.

Ainsi, l'utilisation de Docker constitue un choix stratégique, offrant un environnement stable et modulaire, indispensable pour le développement itératif d'un module Odoo nécessitant des redémarrages fréquents pour la mise à jour des classes et des vues.

## 2 Visual Studio Code

Visual Studio Code (VS Code) est un éditeur de code source extensible et performant, plébiscité pour sa légèreté et son écosystème riche. Il est particulièrement adapté au développement Odoo grâce à sa gestion native des langages clés du framework.

Pour ce projet de gestion des emplois du temps, VS Code a servi d'environnement de développement intégré (IDE) principal. Il a permis l'édition précise des fichiers sources : le Python pour la logique des cours et des créneaux, le XML pour l'architecture des vues (Listes, Formulaire), et le CSV pour les règles de sécurité. Les extensions dédiées à Python et au format XML ont offert des fonctionnalités d'autocomplétion et de détection d'erreurs syntaxiques, accélérant considérablement l'écriture du code.

VS Code a également facilité la navigation dans l'arborescence du module (models, views, security), permettant une gestion claire de la structure des fichiers requise par Odoo. De plus, son terminal intégré a permis d'exécuter directement les commandes Docker nécessaires au

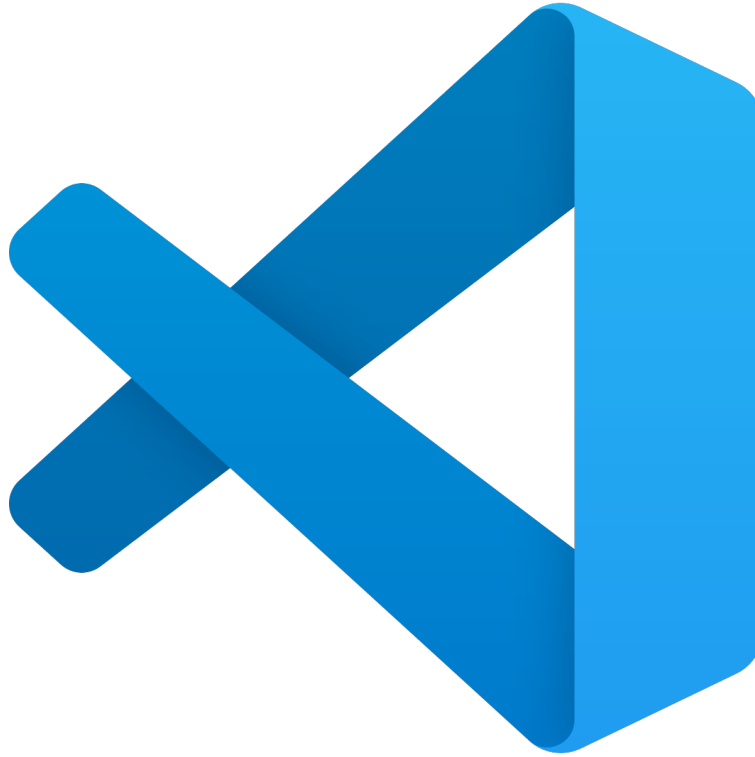


FIGURE 2 – Logo Visual Studio Code

contrôle du cycle de vie de l'application (démarrage, arrêt, logs) sans quitter l'interface de développement.

En somme, Visual Studio Code a offert un cadre de travail ergonomique et unifié, essentiel pour manipuler les différents types de fichiers qui composent notre module de planification.

### 3 Odoo

Odoo est une suite d'applications de gestion d'entreprise open source, réputée pour son architecture modulaire et sa flexibilité. Plus qu'un simple ERP, il fournit un framework de développement complet permettant de créer des applications métier sur mesure en héritant de fonctionnalités natives puissantes.

Dans ce projet, nous avons développé un module personnalisé nommé « Gestion des Emplois du Temps » sur la version 17 d'Odoo. Le choix de cette plateforme s'explique par la puissance de son ORM (Object-Relational Mapping), qui nous a permis de définir nos objets métiers (Cours, Professeurs, Salles) sous forme de classes Python, laissant Odoo gérer automatiquement la création des tables dans la base de données PostgreSQL.

L'architecture MVT (Modèle-Vue-Template) d'Odoo a été exploitée pour séparer la logique de planification de son affichage. Nous avons utilisé le moteur de vues XML pour concevoir des interfaces intuitives permettant de visualiser le planning sous forme de listes et de formulaires détaillés, intégrant des widgets spécifiques comme la barre de statut (pour les états "Confirmé" ou "Annulé").

De plus, le système de sécurité natif d'Odoo a été mis à profit via le fichier `ir.model.access.csv`





FIGURE 3 – Logo Odoo

pour définir précisément les droits de lecture et d’écriture sur les emplois du temps, sécurisant ainsi l’accès aux données sensibles de l’établissement.

Odoo constitue donc le cœur technologique de notre solution, fournissant les briques logicielles nécessaires pour transformer un besoin complexe de planification en une application fluide et intégrée.

## 4 Git & Git Bash

Git Bash est une interface en ligne de commande permettant d’exploiter la puissance du système de contrôle de version Git sous environnement Windows, tout en fournissant des utilitaires Unix essentiels.

Dans le cadre de notre projet, Git Bash a joué un double rôle crucial. D’une part, il a servi d’interface principale pour interagir avec le système de fichiers et exécuter les commandes de création de la structure du module (`mkdir`, `touch`), garantissant le respect de l’arborescence standard d’Odoo. D’autre part, il a été l’outil privilégié pour le versionnage de notre code source.

L’utilisation de Git a permis de suivre l’évolution des fichiers du module (modèles, vues, manifestes) et de conserver un historique des modifications apportées aux fonctionnalités de planification. Cette pratique est indispensable pour sécuriser le développement, permettant de revenir à une version stable en cas d’erreur critique lors de l’implémentation des contraintes métier.

Ainsi, Git Bash a été un compagnon quotidien du développement, assurant à la fois la manipulation efficace des fichiers du projet et la gestion rigoureuse du code source de notre module de gestion des emplois du temps.

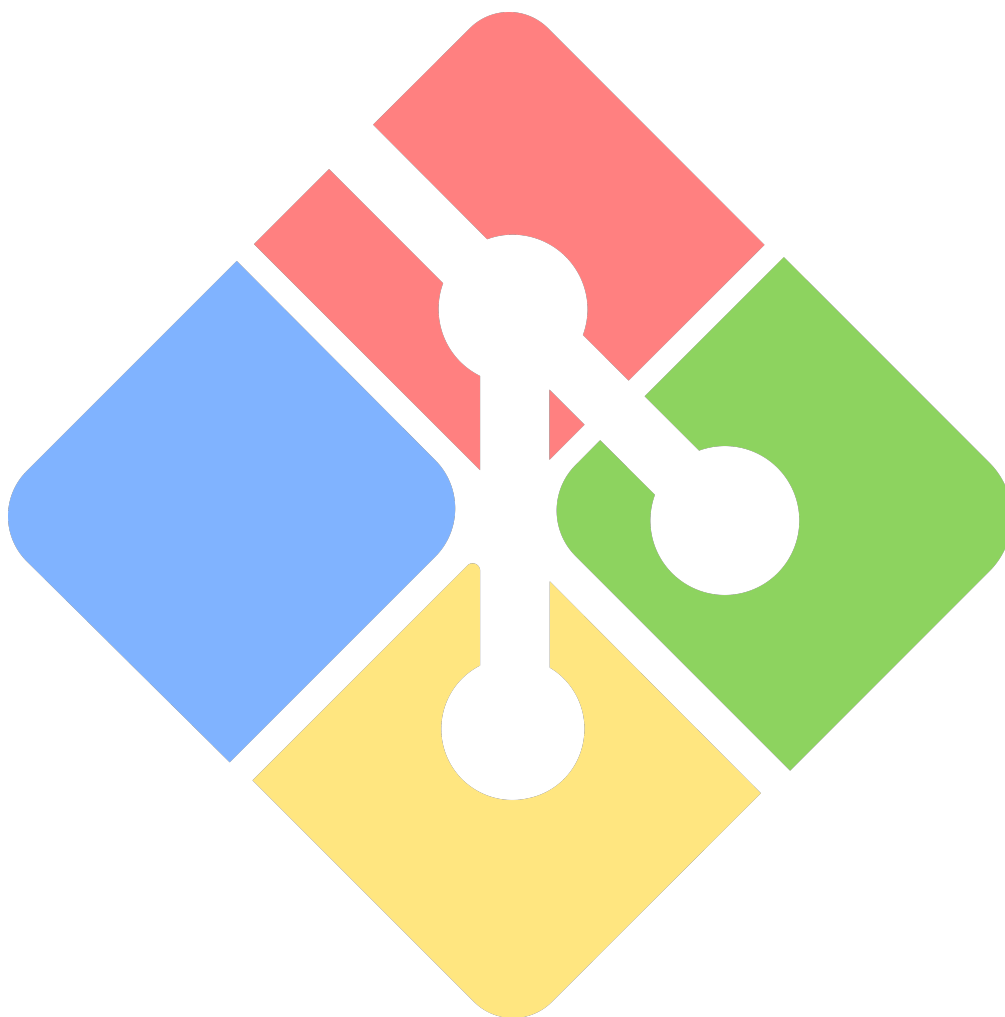


FIGURE 4 – Logo Git &amp; Git Bash

## 5 Python



FIGURE 5 – Logo Python

Python est un langage de programmation interprété, reconnu pour sa syntaxe claire et sa concision. Il est le langage fondamental du serveur Odoo et constitue le moteur de toute la logique métier des applications développées sur cette plateforme.

Dans ce projet, Python a été utilisé pour définir la structure de données de notre module via le fichier `planning.py`. C'est grâce à Python et aux classes fournies par Odoo (`models.Model`) que nous avons pu modéliser les entités « Cours » avec leurs attributs spécifiques : nom de la matière, date, heure de début, durée, et salle.

Au-delà de la simple définition des données, Python a permis d'implémenter l'intelligence du module. Nous l'avons utilisé pour définir des champs calculés, des valeurs par défaut (comme la durée standard d'un cours) et des contraintes de validation. C'est le code Python qui assure, par exemple, la gestion des états du workflow (brouillon, confirmé, terminé) et garantit la cohérence des informations saisies par les utilisateurs. +1

L'intégration native de Python avec l'ORM d'Odoo a grandement simplifié les interactions

avec la base de données, nous permettant de nous concentrer sur les règles de gestion académique plutôt que sur les requêtes SQL brutes. Python est donc la clé de voûte fonctionnelle de notre solution de gestion des emplois du temps.

## Conclusion

Ce chapitre a permis de présenter l’environnement technologique déployé pour le développement du module de « Gestion des Emplois du Temps ». Les outils sélectionnés, à savoir Docker, Visual Studio Code, le framework Odoo et Git Bash, forment un écosystème cohérent et performant.

L’utilisation de Docker a garanti un environnement d’exécution stable et isolé pour le serveur Odoo et la base PostgreSQL. VS Code a offert le confort et la précision nécessaires à l’édition du code Python et XML. Odoo a fourni l’architecture modulaire et les composants métier indispensables à la création d’une application de planification robuste. Enfin, Git Bash a assuré la gestion des versions et la manipulation fluide des fichiers du projet.

La maîtrise de cette stack technique a été déterminante pour la réussite du projet. Elle a permis de structurer le développement selon les standards professionnels et de préparer le terrain pour l’implémentation fonctionnelle.

Le chapitre suivant sera consacré à la réalisation pratique du module. Il détaillera les étapes de conception, de la modélisation des données académiques à la création des interfaces utilisateurs, concrétisant ainsi la solution de gestion des plannings au sein de l’ERP.

## Chapitre 3 : Conception et modélisation

## Introduction

La phase de conception et de modélisation constitue une étape clé dans la réalisation de la plateforme de gestion des emplois du temps et des ressources pédagogiques. Elle permet de traduire les besoins fonctionnels identifiés en une structure logique et cohérente, servant de base au développement de la solution informatique sous Odoo.

Ce chapitre a pour objectif de présenter les choix de conception adoptés pour le projet, en mettant l'accent sur la modélisation des données académiques, la définition des relations entre les différentes entités (cours, enseignants, salles) et la structuration globale du système. Cette phase vise à assurer une organisation claire des informations, à garantir l'intégrité des données (éviter les conflits de planning) et à faciliter la mise en œuvre des fonctionnalités prévues.

La conception repose sur l'analyse des acteurs impliqués et des processus liés à la planification académique, notamment la création des créneaux horaires, l'affectation des professeurs, la réservation des salles et le suivi du statut des cours (confirmé, annulé). Les différents modèles proposés permettent de représenter ces éléments de manière structurée et compréhensible.

Ainsi, ce chapitre présente les différents diagrammes et modèles utilisés pour décrire le fonctionnement de la plateforme, tels que les diagrammes de cas d'utilisation, de séquence et de classes. Il constitue une étape essentielle pour assurer une implémentation efficace, cohérente et évolutive de la solution développée.

## 1 Diagramme de cas d'utilisation

La figure suivante présente le diagramme de cas d'utilisation du système de gestion des emplois du temps développé sous l'ERP Odoo. Ce diagramme met en évidence les différentes interactions entre les acteurs principaux du système, principalement l'administrateur (ou responsable de planning) et secondairement les enseignants ou étudiants (en consultation), ainsi que les fonctionnalités essentielles mises à leur disposition.

Il illustre de manière globale les opérations de planification des cours, l'association des ressources (salles et professeurs) aux créneaux horaires, la gestion des statuts de chaque séance à travers un workflow (brouillon, confirmé, terminé), ainsi que la consultation des disponibilités. Ce diagramme permet ainsi de mieux comprendre le fonctionnement général du système, les responsabilités de l'administrateur dans l'élaboration du planning et la manière dont les différentes fonctionnalités s'articulent pour assurer une gestion structurée et sans conflit des ressources pédagogiques.

## 2 Diagramme de séquence

Le diagramme de séquence constitue un outil essentiel pour décrire le comportement dynamique du système de gestion des emplois du temps. Il permet de représenter de manière chronologique les interactions entre l'administrateur et le système, ainsi que les composants applicatifs impliqués dans le traitement des informations de planification.

À travers ce diagramme, il est possible de suivre les principales étapes du cycle de vie d'un créneau de cours, depuis sa création par l'administrateur, la sélection de la matière, l'assignation de l'enseignant et de la salle, jusqu'à la validation finale du créneau. Ce diagramme met en

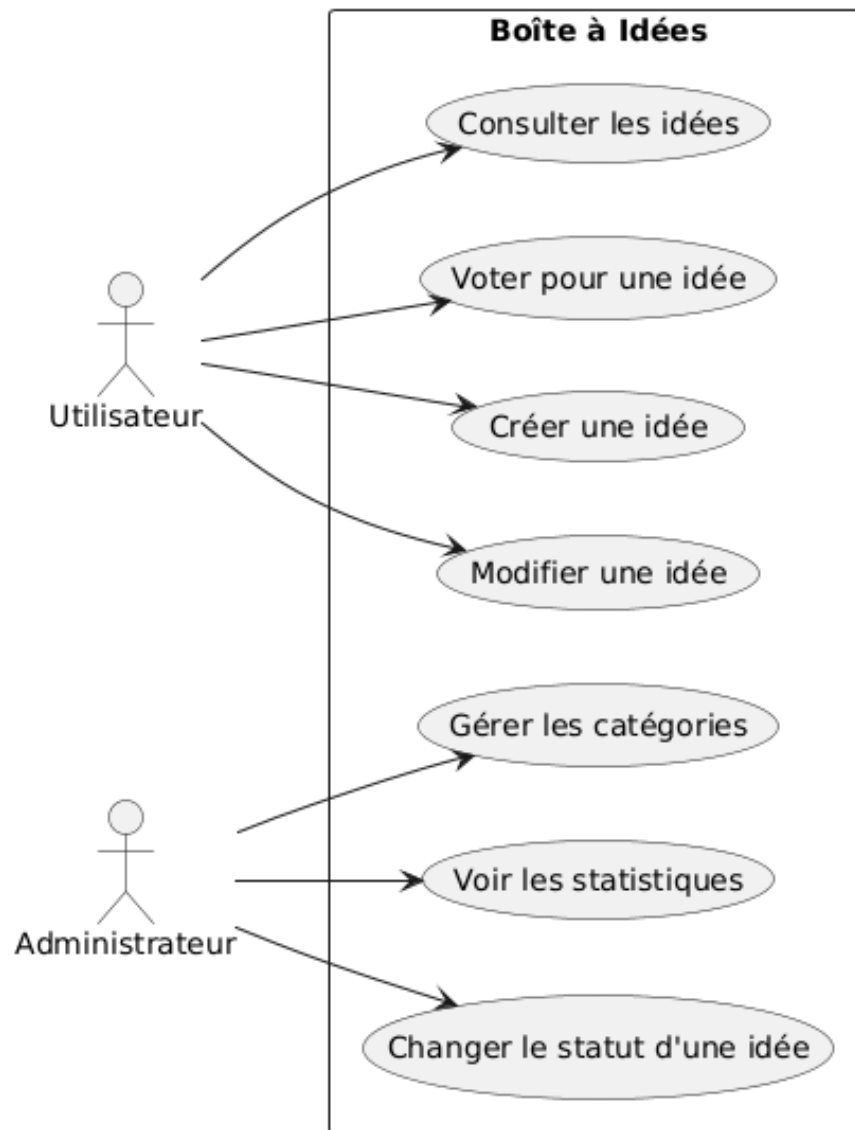


FIGURE 6 – diagramme de cas d'utilisation

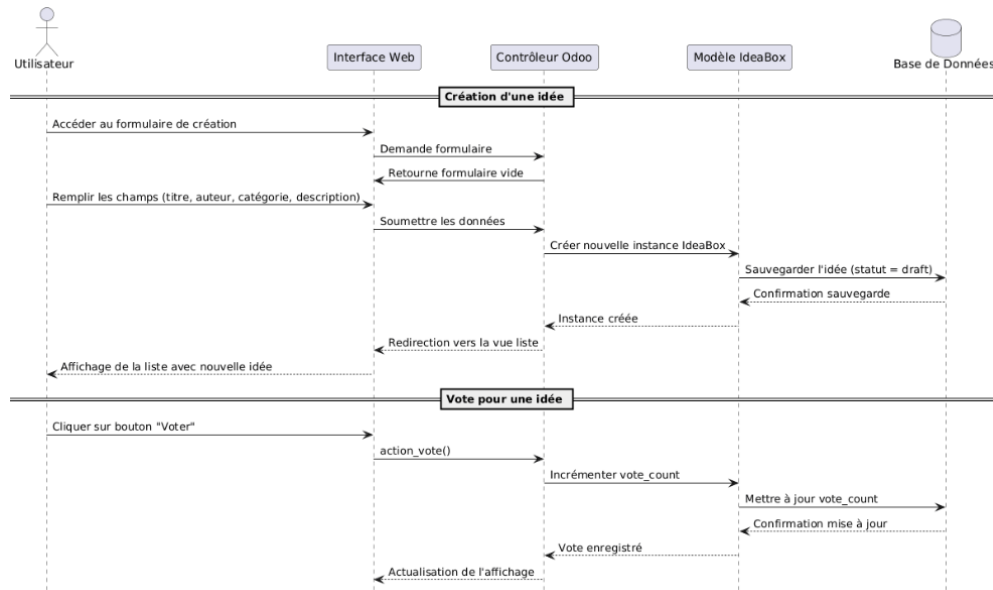


FIGURE 7 – Diagramme de séquence

évidence la manière dont les actions des utilisateurs sont prises en charge par l'application, traitées par le module Odoo (vérification des contraintes horaires) et enregistrées dans la base de données.

Il permet également de mieux comprendre la circulation des informations, notamment lors des changements d'état (ex : passage de "Prévu" à "Confirmé"), tout en assurant une cohérence dans l'enchaînement des opérations. Enfin, le diagramme de séquence facilite l'analyse du fonctionnement global de la plateforme et constitue un support important pour valider les règles métier avant la phase de développement.

### 3 Diagramme de classe

Le diagramme de classes représente la structure statique du système de gestion des emplois du temps en mettant en évidence les principales entités du domaine ainsi que les relations qui existent entre elles. Il permet de modéliser les données manipulées par l'application et d'illustrer l'organisation logique du système.

Dans ce projet, les classes principales sont le Cours (ou Créneau), l'Enseignant et la Salle, qui constituent le cœur du processus de planification. Chaque créneau de cours est associé à une matière spécifique, à un enseignant responsable et à une salle de cours, ce qui permet d'assurer une organisation claire et d'éviter les doubles réservations.

Le diagramme met également en évidence les attributs essentiels tels que la date, l'heure de début, la durée et le statut du cours. De plus, l'utilisation des énumérations pour les types de salles (Amphi, Labo, etc.) et les états du workflow permet de formaliser les contraintes métier. Les relations entre les classes traduisent les règles de gestion du système (ex : un professeur peut avoir plusieurs cours, mais un cours n'a lieu que dans une seule salle à la fois) et garantissent la cohérence des données. Ainsi, ce diagramme de classes constitue une base



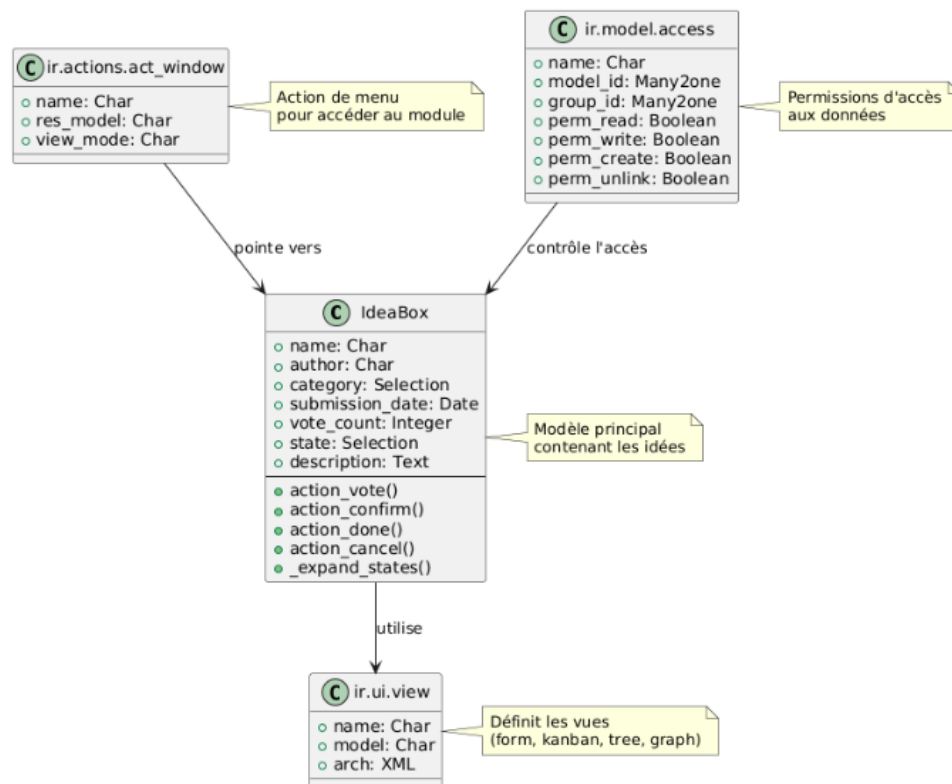


FIGURE 8 – Diagramme de classe

essentielle pour l'implémentation des modèles ORM dans Odoo et pour la compréhension globale de l'architecture du système.

## Conclusion

Ce chapitre a permis de poser les bases conceptuelles et logiques de notre module de gestion des emplois du temps. À travers l'élaboration des diagrammes de cas d'utilisation, de séquence et de classes, nous avons pu traduire les exigences fonctionnelles en une architecture structurée et cohérente.

Cette étape de modélisation a été déterminante pour définir précisément les interactions entre les utilisateurs et le système, ainsi que pour figer la structure des données (Cours, Enseignant, Salle) qui sera implémentée dans la base PostgreSQL. Elle garantit une compréhension claire des règles de gestion, notamment en ce qui concerne l'affectation des ressources et le cycle de vie des créneaux horaires, minimisant ainsi les risques d'erreurs lors du développement.

Disposant désormais d'une vision claire et détaillée de la solution à concevoir, nous pouvons entamer la phase de construction. Le chapitre suivant sera donc consacré à la réalisation technique du projet, détaillant la mise en œuvre effective du module au sein de l'environnement Odoo, de la traduction des modèles en code Python jusqu'à la création des interfaces utilisateurs.

## Chapitre 4 : Réalisation et mise en œuvre

## Introduction

Après avoir modélisé les processus de soumission et de vote des idées dans le chapitre précédent, nous consacrons cette partie à la présentation du résultat final : le module « Boîte à Idées » intégré à l'environnement Odoo.

Ce chapitre expose les différentes interfaces utilisateurs développées pour répondre aux besoins d'ergonomie et d'interactivité identifiés lors de l'analyse. Nous détaillerons les vues principales (Kanban, Liste, Graphique et Formulaire) qui permettent aux collaborateurs de proposer leurs initiatives et aux managers de piloter l'innovation interne de manière visuelle et intuitive. Chaque interface a été conçue pour faciliter l'engagement des utilisateurs et la transparence du processus de décision.

### 1 Vue Kanban - Gestion visuelle du flux d'idées

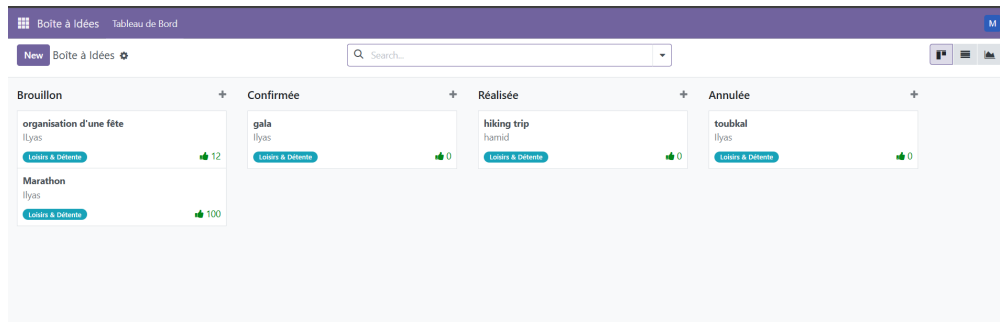


FIGURE 9 – Vue Kanban - Gestion visuelle du flux d'idées

Cette interface représente le cœur de l'expérience utilisateur du module. La vue Kanban organise les idées sous forme de cartes dynamiques, réparties en colonnes selon leur état d'avancement dans le workflow : « Brouillon », « Confirmée », « Réalisée » et « Annulée ». Chaque carte offre un résumé visuel immédiat, affichant le titre de l'idée, l'auteur, la catégorie (ex : Loisirs & Détente) ainsi que le nombre de votes reçus (indiqué par l'icône de pouce levé). Cette vue permet un pilotage agile : un simple « glisser-déposer » suffit pour faire évoluer une idée d'une étape à l'autre, offrant une vision claire du pipeline d'innovation de l'entreprise.

### 2 Vue Liste - Supervision globale et classement

Pour une gestion plus administrative et exhaustive, la vue Liste (ou « Tree View ») présente l'ensemble des propositions sous forme tabulaire. Elle permet de visualiser rapidement les informations clés : le titre, l'auteur, la catégorie, le score de popularité (Votes) et le statut actuel via des badges colorés. Cette interface est particulièrement utile pour comparer les idées entre elles. Un atout majeur de cette vue est sa capacité d'agrégation : comme on peut le voir au bas de la colonne « Votes », le système calcule automatiquement le total des votes (ici 112), permettant de quantifier l'engagement global des collaborateurs.

Titre de l'idée	Auteur	Catégorie	Votes	Statut
<input type="checkbox"/> organisation d'une fête	Ilyas	Loisirs & Détente	12	Brouillon
<input type="checkbox"/> Marathon	Ilyas	Loisirs & Détente	100	Brouillon
<input type="checkbox"/> gala	Ilyas	Loisirs & Détente	0	Confirmée
<input type="checkbox"/> hiking trip	hamid	Loisirs & Détente	0	Réalisée
<input type="checkbox"/> toubkal	Ilyas	Loisirs & Détente	0	Annulée
			112	

FIGURE 10 – Vue Liste - Supervision globale et classement

### 3 Vue Graphique - Analyse statistique des tendances

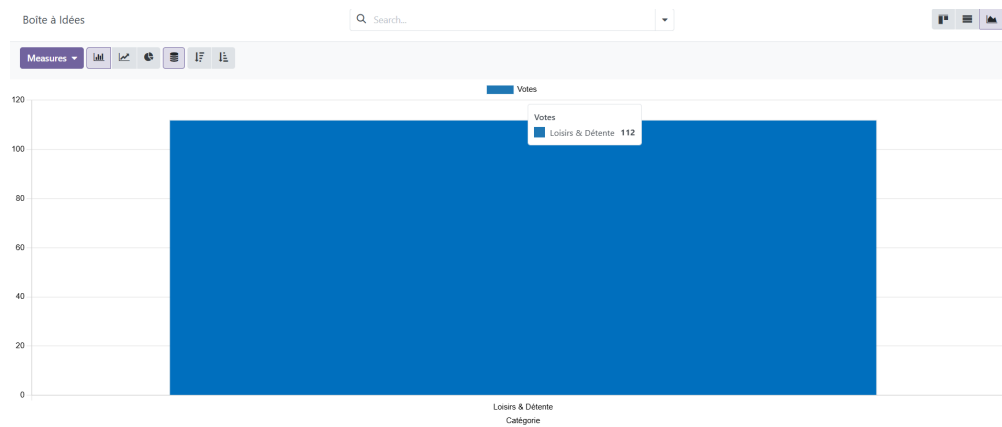


FIGURE 11 – Vue Graphique - Analyse statistique des tendances

Afin d'apporter une dimension décisionnelle au module, nous avons implémenté une vue Graphique native. Cette interface transforme les données brutes en informations exploitables par la direction. L'historique présenté ici illustre la répartition des votes par catégorie (ex : « Loisirs & Détente »). Cet outil d'analyse permet d'identifier en un coup d'œil les thématiques qui suscitent le plus d'intérêt ou de préoccupation au sein des équipes, facilitant ainsi l'orientation des investissements ou des actions prioritaires à mener.

### 4 Vue Formulaire - Détail et cycle de vie de l'idée

La vue Formulaire constitue l'interface de consultation détaillée et d'édition d'une initiative. Elle structure l'information de manière claire : le titre de l'idée, l'auteur, la date de soumission et la catégorie sont mis en évidence, suivis d'une description textuelle libre. L'élément central de cette vue est la barre d'état (Workflow) située en haut à droite, qui indique clairement l'étape actuelle du processus (ici « Annulée »). De plus, l'intégration d'un « Smart Button » (bouton intelligent) affichant le compteur de votes permet de consulter la popularité de l'idée sans surcharger l'interface principale.

The screenshot shows the 'Boîte à Idées' (Idea Box) interface in Odoo. The form is titled 'jour à la plage'. It includes the following fields and values:

- Auteur:** Ilyas
- Date:** 01/23/2026
- Catégorie:** Loisirs & Détente
- Description:** Journée inoubliable

The top navigation bar shows 'Boîte à Idées' and 'Tableau de Bord'. A 'Votes' section shows 8 votes. The right side features a progress bar with stages: 'Brouillon', 'Confirmée', 'Réalisée', and 'Annulée'.

FIGURE 12 – Vue Formulaire - Détail et cycle de vie de l'idée

## Conclusion

Ce chapitre a permis de démontrer la concrétisation technique de notre module « Boîte à Idées ». Les interfaces présentées témoignent d'une intégration réussie au sein de l'ERP Odoo, respectant les standards d'ergonomie modernes.

Nous avons mis en place un système complet alliant la simplicité de saisie (Vue Formulaire), la fluidité de gestion (Vue Kanban) et la puissance d'analyse (Vue Graphique). Ces outils visuels répondent parfaitement à l'objectif initial du projet : transformer la collecte d'idées, souvent fastidieuse, en un processus collaboratif, transparent et engageant pour l'ensemble du personnel.

# Conclusion et perspectives

Le travail réalisé dans le cadre de ce projet a permis de concevoir et de développer un module Odoo fonctionnel dédié à la gestion de l'innovation participative : la « Boîte à Idées ». En suivant une démarche structurée, allant de l'analyse des besoins collaboratifs à l'implémentation technique, nous avons pu répondre à la problématique de la centralisation et de la valorisation des initiatives internes. Le module développé centralise désormais les éléments essentiels du cycle de vie d'une idée : la soumission par les collaborateurs, la catégorisation par thématiques, l'évaluation par les pairs via le système de votes et le suivi opérationnel jusqu'à la réalisation. L'utilisation du framework Odoo (version 17) nous a permis de bénéficier d'un environnement de développement robuste, tirant parti de la puissance de l'ORM pour garantir l'intégrité des données et fluidifier les interactions (notamment via l'incrémentation dynamique des scores). Sur le plan ergonomique, l'application offre une expérience utilisateur moderne et engageante grâce à la diversité des vues implémentées. Au-delà des listes standards, l'intégration de vues visuelles comme le Kanban et les Graphiques permet une gestion intuitive du flux d'innovation et une analyse immédiate des tendances. Les mécanismes de workflow (Brouillon → Confirmé → Réalisé) et l'interactivité des boutons de vote témoignent de l'aspect « collaboratif » et dynamique de la solution. En termes de perspectives, plusieurs évolutions pourraient être envisagées pour transformer ce module en un véritable outil de management de l'innovation : Gamification (Ludification) : Mettre en place un système de récompenses (badges, classements) pour les collaborateurs dont les idées récoltent le plus de votes ou sont mises en œuvre, afin de stimuler la participation sur le long terme. Espace de Discussion (Chatter) : Intégrer les fonctionnalités sociales natives d'Odoo pour permettre aux utilisateurs de commenter les idées, de débattre et d'enrichir les propositions directement sous chaque fiche. Transformation en Projet : Automatiser la création d'une tâche dans le module "Gestion de Projet" d'Odoo dès qu'une idée passe au statut « Confirmée », assurant ainsi une continuité parfaite entre l'idéation et l'exécution. Application Mobile : Optimiser les interfaces pour un usage sur smartphone, permettant aux employés de soumettre une idée (avec photo à l'appui) directement depuis le terrain. Intelligence Artificielle : Utiliser des outils d'analyse sémantique pour détecter automatiquement les idées similaires (doublons) ou suggérer des catégories pertinentes lors de la saisie. Ces perspectives constituent des axes réalistes d'amélioration qui feraient de ce module un levier stratégique majeur pour la culture d'entreprise et l'amélioration continue.

## Webographie

Documentation Odoo

Odoo S.A. Odoo 17 – Documentation officielle (Developer et User). <https://www.odoo.com/documentation>

Odoo S.A. Odoo Development – ORM, Models, Views (Kanban, Graph), Actions. <https://www.odoo.com/>

Docker / Docker Compose

Docker Inc. Docker Documentation (Containerization). <https://docs.docker.com/>

Docker Inc. Docker Compose – Documentation & Reference. <https://docs.docker.com/compose/>

Langages & Base de données

Python Software Foundation. Python 3 Documentation (Syntaxe et Librairies). [https ://docs.python.org/3/](https://docs.python.org/3/)

PostgreSQL Global Development Group. PostgreSQL Documentation. [https ://www.postgresql.org/docs/](https://www.postgresql.org/docs/)  
Outils de Développement

Visual Studio Code. Documentation officielle et Extensions Python/XML. [https ://code.visualstudio.com/docs](https://code.visualstudio.com/docs)

Git SCM. Git Documentation & Git Bash for Windows. [https ://git-scm.com/doc](https://git-scm.com/doc)