**Étapes pour Implémenter la Solution avec la Démarche I.D.M.**

**4 étapes principales :**

1. CIM (Computation Independent Model) : Modèle indépendant du calcul (représentation des besoins métier).
2. PIM (Platform Independent Model) : Modèle indépendant de la plateforme (conception abstraite du système).
3. PSM (Platform Specific Model) : Modèle spécifique à la plateforme (adaptation du modèle aux caractéristiques techniques).
4. Code : Implémentation concrète en code.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Étape** | **Objectif** | **Actions principales** |
| **CIM** (Modèle Indépendant du Calcul) | Analyser les besoins métier sans se soucier des contraintes techniques. | - Identifier les acteurs (utilisateurs, capteurs, LED, barrière).  - Décrire les exigences fonctionnelles (détection, ouverture/fermeture, gestion des LEDs). |
| **PIM** (Modèle Indépendant de la Plateforme) | Concevoir le système de manière abstraite, sans tenir compte du matériel. | - Modéliser les états avec un diagramme d’état (IDLE, VEHICLE\_DETECTED, etc.).  - Définir les classes principales (ParkingManager, VehicleParkingLogic, HardwareController). |
| **PSM** (Modèle Spécifique à la Plateforme) | Adapter la solution aux spécificités matérielles de l’ESP32. | - Sélectionner les GPIOs pour les capteurs et LEDs. - Configurer l’environnement (PlatformIO, Arduino).  - Implémenter les classes en intégrant les fonctionnalités spécifiques de l’ESP32. |
| **Code** | Implémenter la solution en code et la tester. | - Initialiser les composants matériels (broches GPIO, timers). - Implémenter la logique métier (gestion des états).  - Tester et valider les fonctionnalités. |