

# Mise en oeuvre du Cycle en V

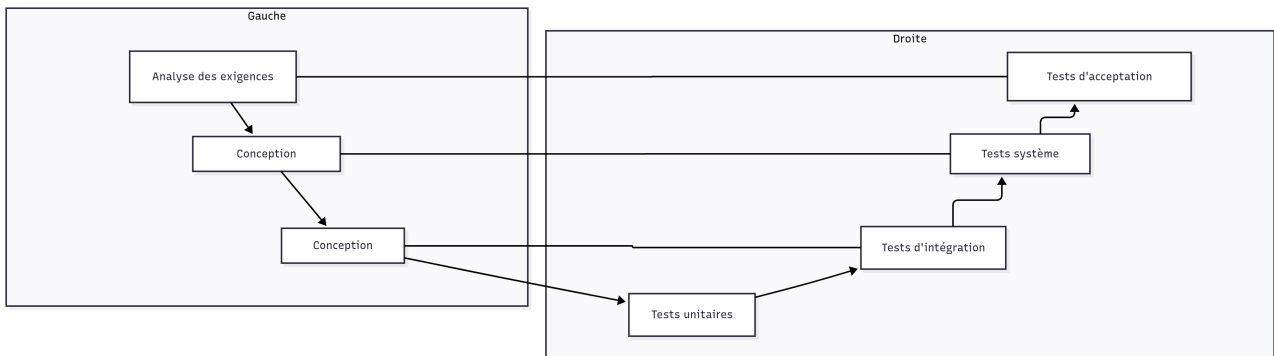
## Mise en oeuvre

Yanis Bennadji

# Introduction

Ce document présente un plan de mise en oeuvre pour un projet de domotique pour une maison connectée structuré selon la méthode de « Cycle en V ».

Nous allons voir comment est géré l'organisation du projet en surface de la définition des exigences, en passant par les tests et jusqu'à la maintenance. Ce document est adressé aux équipes travaillant sur le projet et abordera chaque étape du Cycle en V avec la méthodologie conception  $\leftrightarrow$  validation.



*Cycle en V : correspondance des phases de conception et niveaux de test*

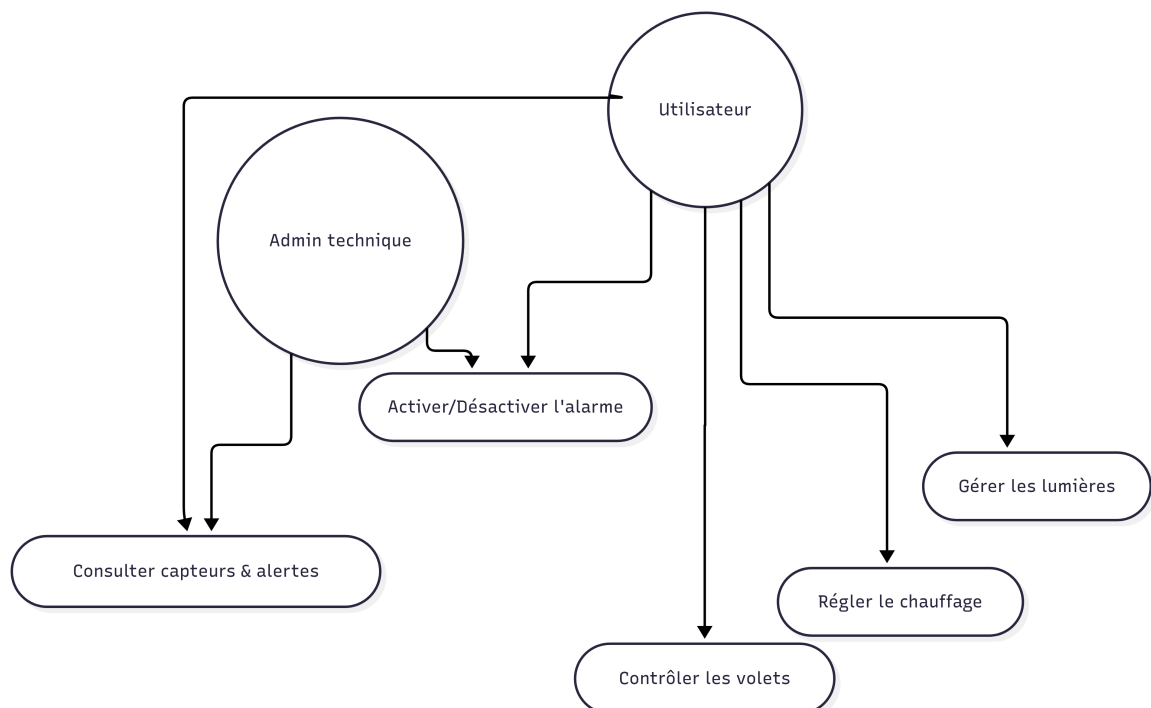
# Spécification des exigences

La spécification des exigences constitue le cahier des charges du projet. Elle permet de définir clairement les fonctionnalités attendues du système domotique ainsi que les contraintes transverses qui guideront sa conception et sa mise en œuvre.

## Portée et objectifs

Le projet vise à développer un système d'automatisation pour une maison connectée, centré sur cinq fonctionnalités de base :

1. Le contrôle des lumières.
2. La gestion du chauffage.
3. La sécurité et l'alarme.
4. Le pilotage des volets.
5. La surveillance par capteurs environnementaux.



Ces fonctions répondent aux besoins essentiels d'un foyer moderne : améliorer le confort, renforcer la sécurité, optimiser la consommation énergétique et fournir une meilleure visibilité sur l'environnement domestique.

### Exigences fonctionnelles :

#### Contrôle des lumières

- L'utilisateur doit pouvoir allumer, éteindre et régler l'intensité des lumières à distance depuis l'application.
- Le système doit permettre de programmer des horaires
- Une détection de présence doit pouvoir déclencher automatiquement l'éclairage.
- L'état des lumières doit être consultable en temps réel.

#### Gestion du chauffage

- L'utilisateur doit pouvoir régler la température pièce par pièce.
- Le système doit exécuter des programmations horaires
- Les capteurs de température doivent envoyer régulièrement leurs mesures.

#### Sécurité et alarme

- L'utilisateur doit pouvoir activer et désactiver le système d'alarme depuis l'application.
- Lorsqu'un capteur de mouvement ou d'ouverture détecte une intrusion, une alerte doit être envoyée en moins de 5 secondes.
- Le système doit activer une sirène et déclencher l'enregistrement vidéo d'une caméra en cas d'intrusion.

#### Contrôle des volets

- L'utilisateur doit pouvoir ouvrir et fermer chaque volet individuellement ou tous simultanément.

- Des scénarios automatiques doivent être configurables

- L'état des volets doit être affiché dans l'application.

#### Capteurs environnementaux

- Le système doit collecter les mesures de capteurs (température, humidité, fumée, CO<sub>2</sub>, inondation).

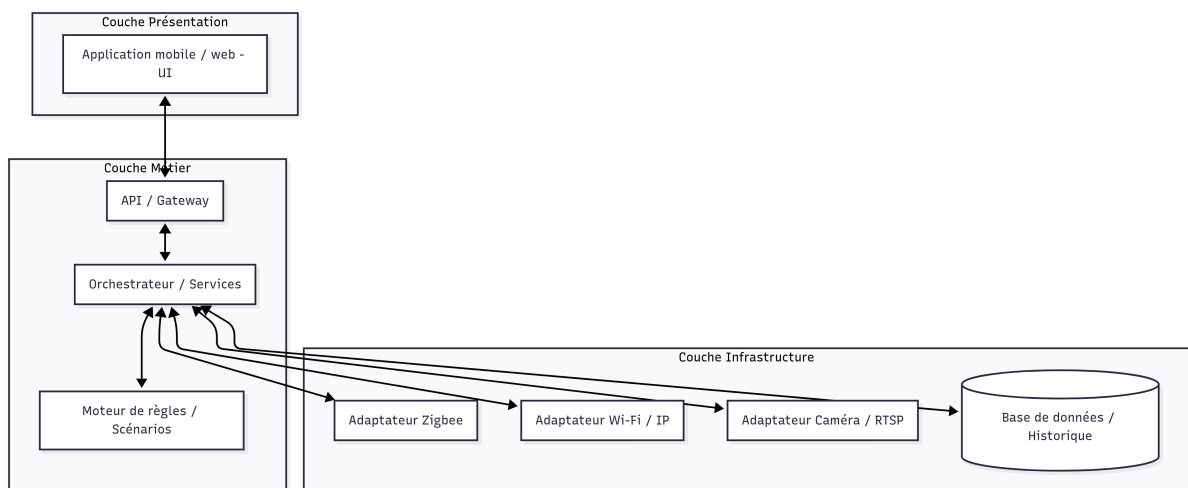
# Conception architecturale et détaillée

## Architecture globale

L'architecture du système repose sur trois couches principales :

- Couche Présentation : l'application mobile et web qui sert d'interface entre l'utilisateur et la maison connectée.
- Couche Métier : le contrôleur central qui exécute la logique, orchestre les scénarios et applique les règles de fonctionnement.
- Couche Infrastructure : les protocoles de communication, les équipements connectés ainsi que la base de données permettant de stocker l'historique et les configurations.

Cette séparation en couches assure la modularité du système et facilite les évolutions futures.



## Composants logiciels

Les composants majeurs identifiés sont :

- Application mobile/web : permet à l'utilisateur de piloter les équipements et de consulter les informations de la maison. L'accent est mis sur la simplicité et l'intuitivité de l'interface.
- API et orchestrateur : expose les services du système et gère la communication entre l'application et les équipements. Il applique également les scénarios automatiques
- Moteur de règles : module responsable de l'exécution des automatisations paramétrées par l'utilisateur
- Base de données : stocke les profils utilisateurs, les historiques de mesures des capteurs, et les journaux d'événements.
- Logiciels embarqués : intégrés aux appareils, ils exécutent des actions simples et communiquent avec le hub via des protocoles adaptés.

## Interfaces utilisateur

L'interface utilisateur doit être claire et accessible pour tout profil d'utilisateur. Elle comprend :

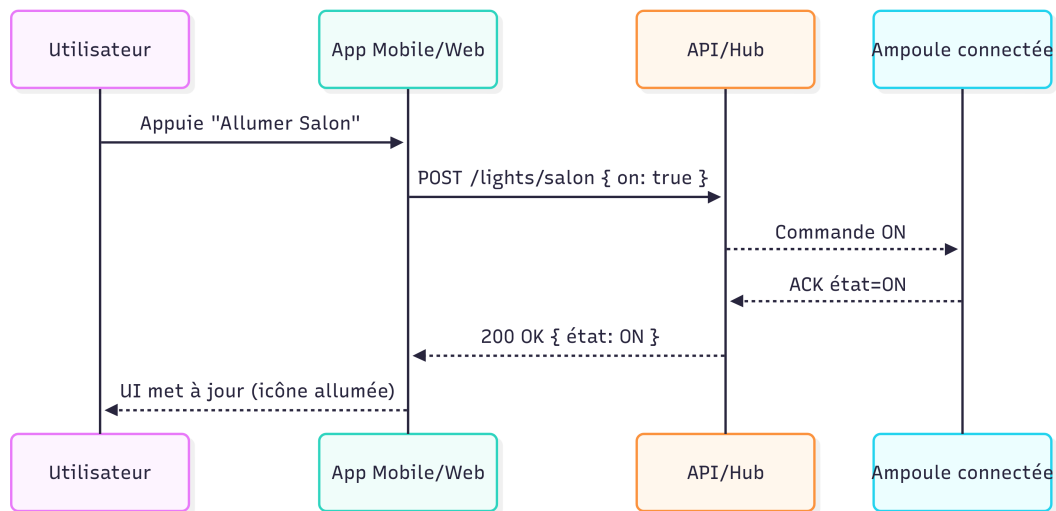
- Un tableau de bord affichant l'état global de la maison
- Un écran de configuration pour définir des scénarios personnalisés
- Un écran de notifications regroupant toutes les alertes et événements.

## Communications et échanges

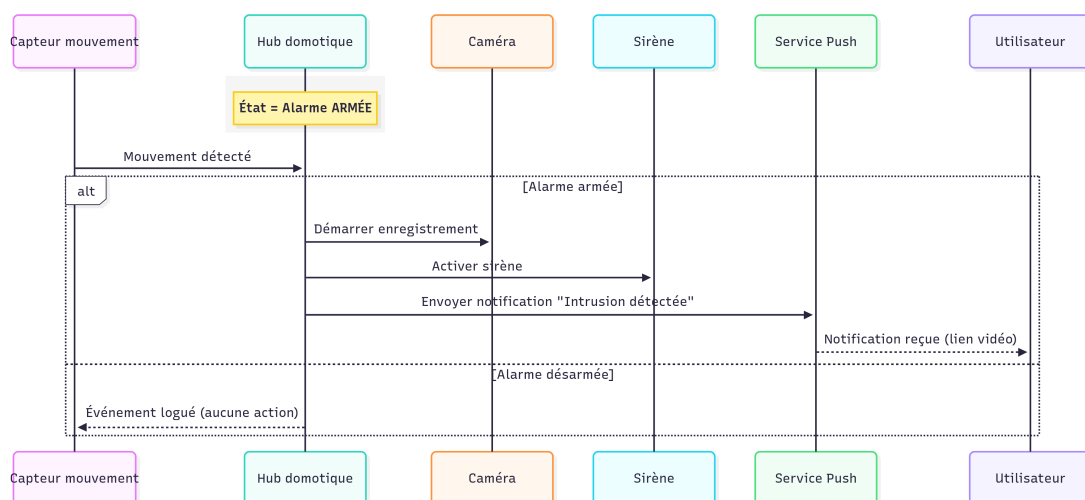
Les échanges de données se font principalement selon deux canaux :

- Sans fil pour relier le hub aux capteurs et actionneurs. Ces protocoles assurent une communication fiable et sécurisée.

- Internet pour accéder au système à distance, recevoir des notifications push ou interagir avec des services externes (API météo, assistants vocaux).



De même, pour la sécurité, lorsqu'un capteur de mouvement détecte une intrusion, le hub active simultanément la sirène, enregistre une vidéo via la caméra et envoie une notification à l'utilisateur.





# Implémentation et tests unitaires

La phase d'implémentation consiste à développer progressivement les modules du système domotique, en respectant l'architecture définie et en validant chaque étape par des tests unitaires.

Chaque fonctionnalité est développée en parallèle par un binôme front-end/back-end et validée dès sa réalisation par l'ingénieur QA.

## Outils et pratiques

- Git pour la gestion du code et la collaboration.
- Intégration continue (CI) afin de lancer automatiquement les tests unitaires.
- Revue de code systématique pour limiter les erreurs.

## Tests unitaires

Les tests unitaires garantissent que chaque module fonctionne de façon isolée avant intégration.

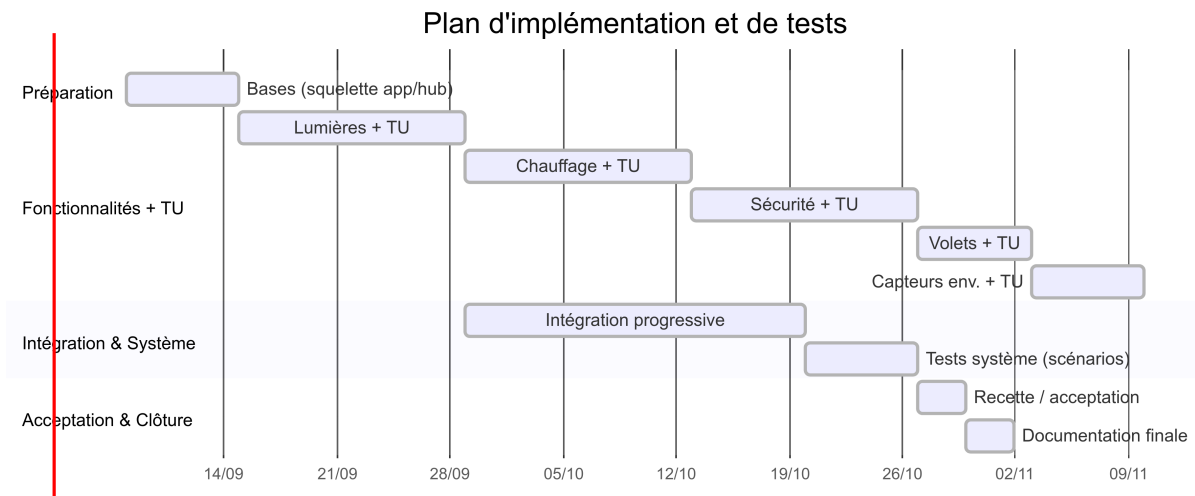
Exemples :

- Vérifier que la fonction `allumerLampe()` modifie bien l'état d'une lampe simulée.
- Vérifier que le calcul de température du module chauffage respecte les consignes.
- Vérifier que l'alarme déclenche une notification lors d'une intrusion.

Un module est considéré comme terminé uniquement si ses tests unitaires passent à 100 %.

## Planning

Le diagramme de Gantt ci-dessous illustre l'enchaînement des phases de développement et de validation. Chaque lot inclut la mise en œuvre et ses tests unitaires.



# Test d'intégration, système et acceptation

La stratégie de tests suit la logique du Cycle en V : chaque exigence définie doit être vérifiée à travers plusieurs niveaux de tests. Ces validations garantissent que le système final répond bien au cahier des charges.

## Tests d'intégration

Ces tests vérifient la bonne interaction entre les différents modules (application, hub et équipements).

- Exemple 1 : depuis l'application, allumer une lumière → vérifier que le hub transmet la commande et que l'ampoule s'allume réellement.
- Exemple 2 : intrusion détectée → vérifier que le hub déclenche la sirène et envoie l'alerte.

## Tests système

Ils portent sur le fonctionnement global de la maison connectée.

- Scénario quotidien : ouverture automatique des volets le matin, activation de l'alarme en journée, chauffage en soirée.
- Scénario d'urgence : détection de fumée → alerte envoyée, volets ouverts automatiquement et chauffage coupé.

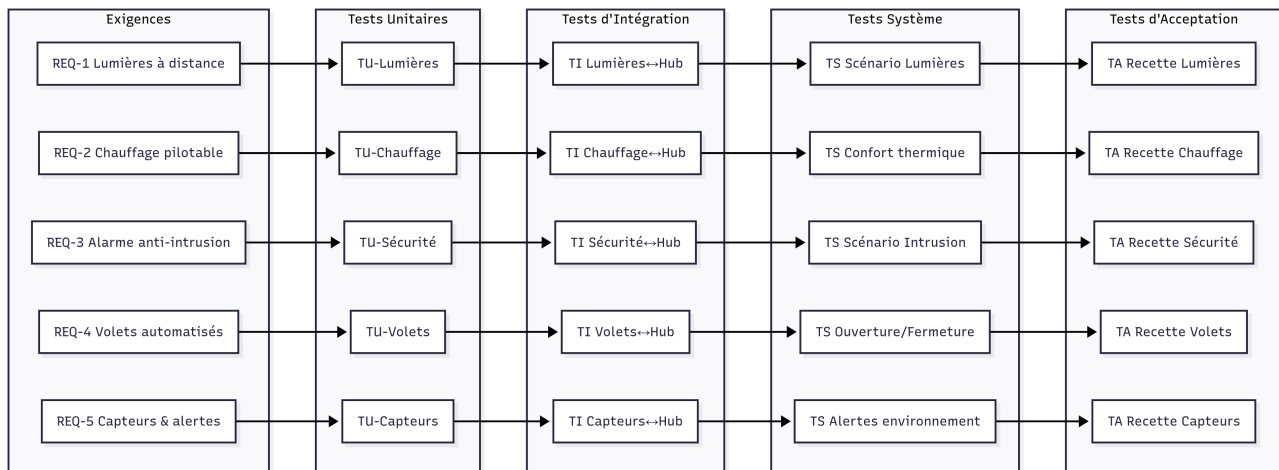
## Tests d'acceptation

Réalisés avec l'utilisateur final, ils valident que chaque exigence du cahier des charges est remplie.

- Exemple : l'utilisateur désarme l'alarme depuis l'application → le système doit refléter immédiatement l'état désactivé.
- Critères d'acceptation : rapidité des actions (alerte < 5 secondes), simplicité d'utilisation, et conformité aux attentes exprimées.

### Traçabilité exigences ↔ tests

Le schéma ci-dessous illustre la correspondance entre chaque exigence (REQ) et les tests unitaires, d'intégration, système et d'acceptation.



# Maintenance et support

## Planification de la maintenance et du support

Une fois le système domotique déployé, il est essentiel d'assurer sa pérennité et sa fiabilité par un plan de maintenance et de support. Cette phase, bien qu'extérieure au développement initial, fait partie intégrante du cycle de vie du projet.

## Maintenance corrective

Malgré les phases de tests, des anomalies peuvent apparaître en production.

- Exemple : un capteur cesse d'envoyer des données ou un volet ne répond plus.
- Solution : mise en place d'un système de suivi des incidents (tickets) et publication de correctifs sous forme de mises à jour logicielles.

## Maintenance évolutive

La maison connectée doit rester adaptable face aux nouveaux besoins.

- Ajout d'équipements (nouveaux capteurs, assistants vocaux).
- Amélioration de l'interface utilisateur ou des scénarios automatiques.

Ces évolutions suivront un processus similaire : analyse des besoins, conception, implémentation et validation.

### Support utilisateur

Un support accessible est prévu pour accompagner les utilisateurs dans l'usage quotidien :

- Fourniture d'un guide utilisateur avec instructions et FAQ.
- Mise en place d'un support technique (email ou hotline) pour résoudre les problèmes majeurs.
- Intégration d'aides contextuelles dans l'application (icônes d'information, tutoriels rapides).

### Suivi et supervision

Le hub domotique collectera des rapports d'état (batterie faible, capteur hors ligne, erreurs). Ces informations permettront une détection préventive des problèmes et faciliteront la maintenance.

### Organisation

La maintenance sera assurée par l'équipe de développement dans un premier temps, puis relayée à une cellule dédiée si le projet devait évoluer. Chaque mise à jour sera accompagnée d'une documentation actualisée et d'une note de version.

# Conclusion

Ce document a présenté le plan de mise en œuvre du projet de maison connectée en suivant la méthodologie du Cycle en V. Chaque étape a été abordée :

- La spécification des exigences a permis d'identifier les fonctionnalités essentielles (lumières, chauffage, sécurité, volets et capteurs) ainsi que les contraintes transverses (sécurité, simplicité, évolutivité).
- La conception architecturale et détaillée a décrit la structure du système, ses composants logiciels, ses interfaces utilisateur et ses échanges de données.
- Le plan d'implémentation a défini l'ordre de développement des fonctionnalités et la mise en place systématique de tests unitaires.
- La stratégie de tests d'intégration, système et acceptation a assuré la traçabilité complète entre exigences et validations.

L'approche en V offre ainsi une vision claire et cohérente, où chaque exigence initiale trouve sa vérification dans une phase de test correspondante. Ce cadre méthodologique assure une meilleure maîtrise des risques, une communication fluide au sein de l'équipe et une livraison conforme aux attentes.