

## Atelier n°1

# Analyse et Refonte d'une Application Universitaire

### Les objectifs :

À l'issue de ce workshop, l'étudiant sera capable de :

- **Analyser une application web existante** développée selon un paradigme procédural, en identifiant son organisation logique et physique.
- **Évaluer de manière critique** les forces et les limites d'une architecture logicielle en termes de maintenabilité, évolutivité, sécurité et lisibilité du code.
- **Identifier les problèmes de conception** freinant l'évolution fonctionnelle et technologique d'un système d'information.
- **Justifier le choix d'un paradigme de programmation et d'une architecture cible** plus adaptés aux besoins d'extension et d'interopérabilité.
- **Concevoir une architecture orientée services**, permettant l'intégration de nouvelles fonctionnalités (examens en ligne, plateforme vidéo, applications mobiles via API).
- **Structurer une solution modulaire** en définissant clairement les responsabilités des composants ou services.
- **Formaliser une proposition d'architecture** à l'aide de schémas (UML ou équivalent) et la communiquer efficacement à l'écrit et à l'oral dans un cadre collaboratif.

## 1. Contexte Général

Vous disposez d'une application web de gestion pédagogique développée en PHP natif (sans utilisation de classes ni d'objets).

L'application actuelle permet :

- L'inscription aux modules,
- La consultation des notes,
- La gestion des emplois du temps,
- La communication interne entre utilisateurs.

L'université souhaite désormais étendre ce système en ajoutant :

- Un module d'examens en ligne,
- Une plateforme de vidéos pédagogiques,
- Une interface mobile permettant l'accès aux principales fonctionnalités via API.

Le projet ainsi les ressources sont bien partagées via <https://github.com/badi3a/SOA-Training/tree/Workshop-1-ProjectStudy>

## 2. Travail demandé

### Étape 1 — Analyse de l'existant

1. Examinez le code source fourni (consulter le guide d'installation pour exécuter le projet) et identifiez le paradigme de programmation utilisé. Les forces et limites de cette approche (lisibilité, maintenabilité, sécurité, évolutivité).
2. Décrivez l'architecture logique (organisation du code, modules, flux principaux) et l'architecture physique.

### Étape 2 — Étude critique

3. Listez les problèmes de conception identifiés dans l'application actuelle.
4. Expliquez pourquoi cette architecture pose des difficultés pour répondre aux nouveaux besoins.

### Étape 3 — Proposition d'évolution

5. Proposez un nouveau paradigme de programmation mieux adapté (à justifier).
6. Définissez une nouvelle architecture logique et physique du système avec un découpage des responsabilités en modules ou services et une organisation cible permettant l'intégration du mobile et des nouvelles fonctionnalités.
7. Schématissez votre proposition (diagramme d'architecture ...).

## 3. Livrables attendus

1. Chaque équipe (3-4-étudiants) Un rapport de 4 à 6 pages comprenant :
  - L'architecture actuelle (logique + physique).
  - Les limites et problèmes identifiés.
  - Votre proposition de nouvelle architecture (logique + physique).
  - Le paradigme de programmation et les technologies recommandées et ses avantages.
2. Un schéma UML ou équivalent illustrant votre vision.
3. Une présentation orale courte (5–10 min) en groupe.