Rapport – Laboratoire 1

Nom : Code permanent :

Cours : LOG430 – Architecture Logicielle Session E´t´e 2025

# Informations g´en´erales

* **URL du d´epˆot GitHub/GitLab** :
* **Langage utilis´e** :
* **Base de donn´ees utilis´ee** :
* **ORM utilis´e** :

# Auto-´evaluation par section

* 1. **Structuration du d´epˆot**

|  |  |  |  |
| --- | --- | --- | --- |
| **E´l´ements** | Oui | Non | Commentaire / Justi- fication |
| Le fichier README.md est pr´esent et complet |  |  |  |
| Un fichier .gitignore pertinent est utilis´e |  |  |  |
| Le d´epoˆt pr´esente une structure claire et logique |  |  |  |

* 1. **Application client**

|  |  |  |  |
| --- | --- | --- | --- |
| **E´l´ements** | Oui | Non | Commentaire / Justi- fication |
| Elle permet de rechercher, ajouter, consulter, retourner  des produits |  |  |  |
| L’interface est claire et fonctionnelle |  |  |  |

* 1. **Persistance et ORM**

|  |  |  |  |
| --- | --- | --- | --- |
| **E´l´ements** | Oui | Non | Commentaire / Justi- fication |
| Un ORM est utilis´e pour g´erer la persistance |  |  |  |
| L’application acc`ede correctement a` la base de donn´ees |  |  |  |
| Les op´erations de lecture et d’´ecriture sont persist´ees |  |  |  |
| La base de donn´ees est coh´erente apr`es plusieurs  op´erations concurrentes |  |  |  |
| L’utilisation des indexes dans la base de donn´ees est bien  appliqu´e et justifi´e |  |  |  |

* 1. **Justifications et documentation**

|  |  |  |  |
| --- | --- | --- | --- |
| **E´l´ements** | Oui | Non | Commentaire / Justi- fication |
| Au moins deux d´ecisions architecturales (ADR) sont  r´edig´ees |  |  |  |
| Les choix technologiques sont document´es |  |  |  |
| Les diagrammes UML sont pr´esents (classes, s´equence,  d´eploiement) |  |  |  |
| Les diagrammes sont faits avec un outil de diagramme  comme code (diagram as code), tel que PlantUML ou Mermaid |  |  |  |
| La documentation est ´ecrite en Markdown |  |  |  |
| La documentation suit un gabarit comme Arc42 |  |  |  |
| La documentation est organis´ee dans un r´epertoire d´edi´e  comme “docs” dans le r´epertoire du syst`eme |  |  |  |

* 1. **Conteneurisation et ex´ecution**

|  |  |  |  |
| --- | --- | --- | --- |
| **E´l´ements** | Oui | Non | Commentaire / Justi- fication |
| L’application est conteneuris´ee avec un Dockerfile |  |  |  |
| Cliente est conteneuris´e |  |  |  |
| La base de don´ees instanci´ee manuellement |  |  |  |
| La base de don´ees instanci´ee avec syst`eme d’orchestra-  tion (ex. Docker compose) |  |  |  |
| Les instructions d’ex´ecution sont pr´esentes dans le  README |  |  |  |

* 1. **Int´egration continue (CI/CD)**

|  |  |  |  |
| --- | --- | --- | --- |
| **E´l´ements** | Oui | Non | Commentaire / Justi- fication |
| Une pipeline CI/CD est configur´ee sur GitHub Actions  ou GitLab CI |  |  |  |
| Le **lint** est fonctionnelle |  |  |  |
| L’´etape **test unitaire** est fonctionnelle |  |  |  |
| L’´etape **build Docker** est fonctionnelle |  |  |  |
| L’image est publi´ee sur Docker Hub |  |  |  |

# R´eflexion personnelle

1. **Quels sont les limitations du style d’architecture client/serveur `a 2 couches (2-tier) ?**
2. **Quels sont les limites du m´ecanisme de base de donn´ees utilis´ee ? Quels sont les tactiques pour augmenter la capacit´e de r´eponses (par exemple, r´eduire le temps de r´eponse ou augmenter la quantit´e des usager simultan´es) ?**
3. **Dans le contexte de votre projet du laboratoire 1, quels sont les efforts techniques et les risques potentiels li´es `a la migration du syst`eme de persistance d’une base de donn´ees relationnelle vers une base NoSQL, ou l’inverse ?**
4. **Quels ´el´ements vous ont pos´e le plus de difficult´e ?**
5. **Que souhaitez-vous am´eliorer pour les prochains laboratoires ?**