

Grille de correction détaillée – EC03

Informations générales

- **Épreuve** : EC03 – Développement back-end avec BDD relationnelle + NoSQL
- **Durée** : 4h – Individuelle
- **Projet fil rouge** : RebootCamp – Gestion des utilisateurs, défis, scores et progression
- **Nature** : Épreuve écrite et pratique (POO, MVC, SQL + NoSQL, Docker)

Critères d'évaluation et attentes

Critère	Intitulé officiel	Attentes côté correcteur	Points de vigilance
C7.1	Environnement de dev (Docker, terminal)	Docker-compose fonctionnel, .env propre, commandes reproductibles (logs, migrations).	Vérifier que tout est exécutable rapidement. Attention aux dépendances manquantes.
C8.1	POO et MVC	Architecture claire avec séparation contrôleurs / modèles / vues (ou services). Nommage correct, code orienté objet.	Surveiller les projets qui contiennent toute la logique dans les contrôleurs.
C9	Sécurité et optimisation serveur	Variables d'environnement sécurisées, headers HTTP configurés, gestion des rôles utilisateurs.	Vérifier la bonne séparation admin/utilisateur et l'absence d'expositions sensibles.

Critère	Intitulé officiel	Attentes côté correcteur	Points de vigilance
C10	Base relationnelle (PostgreSQL)	Entités/migrations présentes et cohérentes (User , Challenge). Données test fournies.	Attention aux schémas incomplets ou incohérents avec la logique métier.
C11	Base NoSQL (MongoDB)	Collections utilisées pour progressions, historiques, badges. JSON dump fourni.	Vérifier que Mongo n'est pas utilisé comme simple duplication SQL.
C12	Justification des choix	README clair, arguments sur l'usage SQL vs NoSQL.	Ne pas valider un README vide ou trop générique.
C13	Sauvegarde/restauration	Scripts backup.sh et restore.sh fonctionnels.	Tester la cohérence des dumps (SQL + JSON).

Livrables à corriger

- **Code source** : projet MVC/POO avec back-end complet
- **Dump SQL et JSON** : cohérents, testables, bien documentés
- **Scripts de sauvegarde/restauration** : simples, exécutables, reproductibles
- **README** : expliquant choix techniques, arbitrages SQL vs NoSQL, commandes terminal

Barème indicatif (sur 20 points)

Axe évalué	Points
Environnement et scripts (C7.1, C13)	/4
POO / MVC (C8.1)	/4
Sécurisation / optimisation (C9)	/3

Axe évalué	Points
PostgreSQL (C10)	/3
MongoDB (C11)	/3
Justification des choix (C12)	/3

Tolérance : ± 2 points selon qualité du code, clarté des dumps et pertinence des justifications.

Points positifs attendus

- Entité `User` (avec rôles) et `Challenge` bien modélisées dans PostgreSQL
- Collection Mongo pour la progression (score cumulé, badges, historique des défis)
- Route protégée (JWT) retournant les défis + progression combinée d'un utilisateur
- Docker-compose fonctionnel et clair, avec instructions dans le README
- Scripts de sauvegarde/restauration utilisables directement
- README expliquant pourquoi tel type de données est dans SQL (structuré, relations) et tel autre dans NoSQL (flexible, historique, analytics)

Erreurs fréquentes à surveiller

- Projet mono-BDD (candidat ignore Mongo ou SQL \rightarrow non conforme au sujet)
- Sécurité minimale ou absente : accès non protégés, `.env` exposant mots de passe
- Architecture MVC non respectée (logique dans routes, pas de services ou modèles)
- Dumps incohérents : données inutilisables, structure manquante
- README superficiel, sans justification des arbitrages SQL/NoSQL
- Scripts `backup.sh` / `restore.sh` incomplets ou non exécutables

Rappel pédagogique

Cette épreuve doit confirmer que l'apprenant maîtrise :

- La **mise en place d'un back-end structuré en POO/MVC**
- L'**utilisation combinée** de bases relationnelles et NoSQL, avec une logique claire de répartition des données
- La sécurisation d'un serveur applicatif et la gestion des environnements (Docker, .env)
- La documentation et la justification de **choix techniques** dans un contexte métier réaliste (gamification, progression, classement)
- Les bonnes pratiques de **sauvegarde et restauration** des données

Elle prépare directement à la conception d'APIs sécurisées (EC04) et à l'industrialisation (EC06).