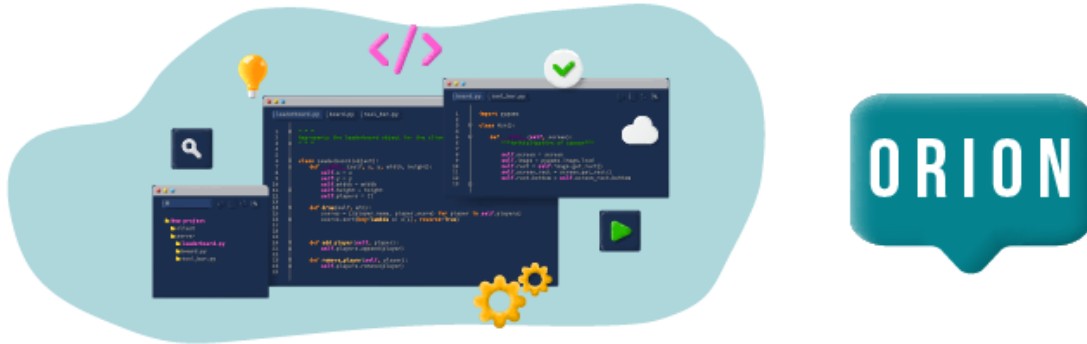


# ORION



## Justification des choix techniques ***Projet MDD***



Auteur : M Pedraza  
Version 0.0.1

## Table des matières

### Aperçu / Synthèse

#### Choix techniques

- Choix 001 : Angular CLI
- Choix 002 : TypeScript
- Choix 003 : Spring Boot
- Choix 004 : Spring Security + JWT
- Choix 005 : MySQL
- Choix 006 : Spring Data JPA
- Choix 007 : Angular Material

## Aperçu / Synthèse

*Ce projet met en œuvre une application full-stack conforme aux contraintes techniques imposées : un frontend développé avec Angular et TypeScript, et un backend en Java basé sur Spring Boot et ses modules (Spring Security, Spring Data JPA). Les échanges entre les deux couches sont sécurisés via JWT, et les données sont stockées dans une base MySQL.*

*Les choix technologiques visent à garantir simplicité, maintenabilité et rapidité de développement dans un contexte MVP.*

- *Frontend : Angular CLI structure le projet et facilite la génération de code. Angular Material fournit une interface moderne et responsive, réduisant l'investissement nécessaire en design CSS.*
- *Backend : Spring Boot permet une configuration rapide et modulaire. Spring Security, couplé à JWT, assure une authentification stateless adaptée aux API REST.*
- *Base de données : MySQL, combiné à Spring Data JPA, offre une gestion*

*fiable et performante des données relationnelles, avec un code SQL minimal.*

*Ces décisions techniques permettent de livrer une application fonctionnelle, évolutive et conforme aux standards reconnus des architectures web, tout en respectant les délais et l'objectif pédagogique du projet.*

## Choix techniques

### Choix 001

choix technique	lien vers le site / la documentation / une ressource	but du choix
Angular CLI	<a href="https://angular.io/cli">https://angular.io/cli</a>	Développement du frontend

*Justification du choix technique : Angular CLI facilite la génération de composants, de services et de modules. Il impose une structure claire au projet et permet un développement rapide tout en respectant les bonnes pratiques Angular. Facilite la génération de composants, services et modules, tout en imposant une structure claire au projet. Permet un développement rapide conforme aux bonnes pratiques Angular.*

- **Avantages :**
  - Standard officiel d'Angular.
  - Génération rapide et automatisée du code.
  - Support intégré pour les tests et le déploiement.
- **Inconvénients :**
  - Nécessite une bonne maîtrise de la ligne de commande.
  - Peut générer du code inutile si mal configuré.

### Choix 002

choix technique	lien vers le site / la documentation / une ressource	but du choix
TypeScript	<a href="https://www.typescriptlang.org/">https://www.typescriptlang.org/</a>	Langage de développement frontend

*Justification du choix technique : TypeScript permet une programmation typée, facilitant la maintenance du code et réduisant les erreurs en développement. Il est parfaitement intégré à Angular. Offre un typage statique qui facilite la maintenance et réduit les erreurs. Parfaitement intégré à Angular.*

**Avantages :**

- *Détection d'erreurs à la compilation.*
- *Lisibilité accrue du code.*
- *Large adoption dans l'écosystème Angular.*

**Inconvénients :**

- *Courbe d'apprentissage pour les débutants en typage strict.*
- *Nécessite une compilation avant exécution.*

Choix 003

choix technique	lien vers le site / la documentation / une ressource	but du choix
Spring Boot	<a href="https://spring.io/projects/spring-boot">https://spring.io/projects/spring-boot</a>	Création de l'API REST backend

*Justification du choix technique : Spring Boot simplifie la configuration d'une application Spring. Il permet de démarrer rapidement avec des configurations par défaut, tout en gardant une grande flexibilité. Il est recommandé par les consignes.*

**Avantages :**

- *Démarrage rapide d'un projet.*
- *Intégration native avec Spring Data et Spring Security.*
- *Documentation riche.*

**Inconvénients :**

- *Surcharge mémoire pour les petites applications.*
- *Courbe d'apprentissage initiale.*

## Choix 004

choix technique	lien vers le site / la documentation / une ressource	but du choix
Spring Security + JWT	<a href="https://spring.io/projects/spring-security">https://spring.io/projects/spring-security</a> <a href="https://jwt.io">https://jwt.io</a>	Authentification et sécurisation des échanges entre frontend et backend

*Justification du choix technique : Spring Security est une solution éprouvée pour la gestion de la sécurité. L'utilisation de JWT permet une authentification stateless adaptée aux APIs REST, avec un bon équilibre entre sécurité et performance.*

### **Avantages :**

- Sécurité robuste et configurable.
- JWT léger et performant pour les échanges HTTP.
- Adapté aux architectures distribuées.

### **Inconvénients :**

- Gestion de l'expiration et du rafraîchissement des tokens à implémenter.
- Nécessite un stockage sécurisé de la clé secrète.

## Choix 005

choix technique	lien vers le site / la documentation / une ressource	but du choix
MySQL	<a href="https://www.mysql.com/">https://www.mysql.com/</a>	Stockage des données utilisateur, articles, commentaires, etc.

*Justification du choix technique : MySQL est une base relationnelle robuste, bien documentée, facile à intégrer avec Spring Data JPA. Elle est gratuite et adaptée à une architecture MVP.*

### **Avantages :**

- Très répandue et largement supportée.
- Bonne performance pour les données relationnelles.

- Outils de gestion graphiques disponibles (ex : MySQL Workbench).

**Inconvénients :**

- Moins adapté aux données non structurées.
- Performances limitées sur des volumes massifs par rapport à certaines alternatives.

Choix 006

choix technique	lien vers le site / la documentation / une ressource	but du choix
Spring Data JPA	<a href="https://spring.io/projects/spring-data-jpa">https://spring.io/projects/spring-data-jpa</a>	Accès aux données

*Justification du choix technique : Permet une gestion simplifiée de la persistance grâce aux repositories. Réduit la quantité de code SQL à écrire tout en assurant des performances correctes.*

**Avantages :**

- Simplification du code d'accès aux données.
- Support des requêtes personnalisées.
- Intégration directe avec Hibernate.

**Inconvénients :**

- Abstraction pouvant masquer des problèmes de performance SQL.
- Courbe d'apprentissage pour les annotations avancées.

Choix 007

choix technique	lien vers le site / la documentation / une ressource	but du choix
Angular Material	<a href="https://material.angular">https://material.angular</a>	Mise en forme et design de

choix technique	lien vers le site / la documentation / une ressource	but du choix
	<a href="#"><u>io/</u></a>	<i>l'interface utilisateur frontend</i>

*Justification du choix technique : Angular Material fournit une bibliothèque de composants UI modernes, accessibles et responsives, entièrement intégrée à Angular. Elle respecte les standards Material Design de Google, permettant un rendu professionnel sans investissement excessif dans le CSS. Idéal pour un MVP clair et fonctionnel.*

**Avantages :**

- Composants préconçus et responsives.
- Gain de temps sur le design CSS.
- Accessibilité intégrée.

**Inconvénients :**

- Augmentation de la taille du bundle.
- Style graphique standard pouvant manquer de personnalisation.