



## Projet Final

# Projet Final - Plateforme Multiservices DeliveCROUS



## Objectif

**Plateforme DeliveCROUS** - 3 modules avec des modèles de données différents :

1. **Module Livraison** (gestion commandes + logistique)
2. **Module Service Client** (tickets support + historique)
3. **Module Recommandations** (notations + suggestions)

### Mission :

- Choisir un **modèle de données** adapté pour **chaque module** (les 3)
- **Justifier techniquement** avec arguments solides, benchmarks, trade-offs
- Définir l'**architecture complète** (schémas, interopérabilité, formats, flux)
- *Optionnel* : Implémenter 1 module pour valider vos choix

### Évaluation :

- **50%** Justification choix de modèles (les 3 modules, comparatifs détaillés)
- **30%** Architecture et interopérabilité (schémas, diagrammes, formats)
- **20%** Qualité du rapport et analyse critique

### Bonus (jusqu'à +20%) :

- Implémentation d'1 module complet (+15%)
- Features avancées (tests, déploiement, etc.) (+5%)

## Les 3 Modules

### Module 1 : Livraison 🚚

- **Entités** : Users, Plats, Commandes, PointsLivraison
- **Enjeux** : Transactions critiques, relations strictes, intégrité données
- **Choix** : Relationnel ou Document ?

### Module 2 : Service Client 💬

- **Entités** : Tickets, Messages, Historique
- **Enjeux** : Schéma flexible, gros volume, recherche full-text
- **Choix** : Document ou Relationnel ?

### Module 3 : Recommandations ⭐

- **Entités** : Notations, Préférences, Relations User↔Plat
- **Enjeux** : Traversées de graphe, patterns collaboratifs
- **Choix** : Graphe, Document ou Relationnel ?

---

## Interopérabilité

**Formats** : JSON (API REST), XML (legacy/SOAP), YAML (config), CSV (exports)

**Flux** :

- Livraison → Service Client (tickets)
- Livraison → Recommandations (historique)
- Recommandations → Livraison (suggestions)

**Projet personnalisé ?** Email `n5sql@sidali.dev` pour validation



# Rendu

**Date :** Dimanche 7 Décembre 2025 à 23h59 **Équipes :** Binômes

**Email :** `n5sq1@sidali.dev` **Objet :** `[NODE.JS] Projet Final Multiservices - GROUPE X`

## Contenu :

- Noms + rôles
  - Rapport PDF (10-12 pages)
  - *Si code* : Lien GitHub + module implémenté
- 



## Rapport (10-12 pages)

### 1. Contexte (1-2 pages)

- Présentation + cas d'usage

### 2. Justification 3 Modules (5-6 pages) - 50% note

- Pour chaque module : choix + arguments + contre-arguments + trade-offs
- Tableau comparatif
- Benchmarks/estimations

### 3. Architecture (3-4 pages) - 30% note

- Schémas de données (ERD/document/graphe)
- Diagramme d'architecture
- Interopérabilité (JSON/XML/YAML/CSV)
- Flux de communication

### 4. Implémentation (*si code*) (2-3 pages)

- Architecture code
- Captures requêtes

- Sécurité

## 5. Analyse Critique (1-2 pages) - 20% note

- Limites des choix
- Alternatives
- Évolutions



## Code (Optionnel - Bonus +15%)

### Base (+10%) :

- Auth JWT + CRUD + 1 relation + validation

### Complet (+15%) :

- Refresh tokens + pagination + 2 relations + feature avancée (tests/déploiement/WebSockets/search/cache)

### Extra (+5%) :

- CI/CD, Swagger, rate limiting



## Aide au Choix : SQL vs NoSQL

Critère	SQL	NoSQL
Transactions	✓ ACID natif	⚠ Limité
Relations	✓ Foreign Keys	Références manuelles
Flexibilité schema	Migrations	✓ Flexible
Scaling	Vertical	✓ Horizontal

**SQL** → Relations fortes, transactions critiques (ex: e-commerce, banque) **NoSQL** → Schéma flexible, scaling horizontal (ex: réseaux sociaux, logs)

---

## Checklist

### Rapport :

- ☐ Justification 3 modules (5-6 pages)
- ☐ Tableau comparatif
- ☐ Schémas données + diagramme architecture
- ☐ Interopérabilité (JSON/XML/YAML/CSV)
- ☐ Analyse critique
- ☐ PDF 10-12 pages

### Code (optionnel) :

- ☐ Auth JWT + CRUD + validation
  - ☐ Relations + README + GitHub
  - ☐ *Pour +15%* : features avancées
- 

## Ressources

### Cours :

- [Cheatsheet](#)
- [SQL vs NoSQL](#)
- [Tutoriel SQL \(Prisma + SQLite\)](#)
- [Tutoriel NoSQL \(Mongoose + NeDB\)](#)

### Maquettes :

- [Figma DeliveCROUS](#)

Questions ? → ``n5sql@sidali.dev``

---

Précédent

← Cas d'utilisation des WebSockets

Fait avec amour ❤ par [\*\*sidali.dev\*\*](#)