

# **CAHIER DES CHARGES**

## **PROJET DE SESSION**

### **NeoSanté RAG**

Plateforme de télémédecine multimodale intégrant l'IA  
RAG et l'analyse de données médicales

**RÉALISER UN CONTRAT DE SERVICE WEB  
TRANSACTIONNEL**

**Réalisé par :**

Sandra Verónica Soto Polo  
Juan Camilo Beltrán Dueñas

**Enseignant : Steve Ataky**

**Institut Teccart – Group 1002 – Automne 2025**

## **1. Contexte et justification**

Dans le contexte post-pandémique, la télémédecine s'impose comme un pilier essentiel des systèmes de santé modernes.

Les besoins croissants en téléconsultation sécurisée, en analyse intelligente de dossiers médicaux et en outils d'aide à la décision clinique ont mis en évidence les limites des plateformes monolithiques actuelles.

Le projet NeoSanté RAG vise à créer une application web innovante qui combine intelligence artificielle, reconnaissance multimodale (texte et image) et architecture microservices, facilitant la consultation à distance et l'aide au diagnostic clinique.

## **3. Objectifs de l'application**

### **Objectifs généraux :**

- Développer une plateforme de télémédecine sécurisée et modulaire.
- Intégrer une IA RAG (Retrieval-Augmented Generation) pour assister le diagnostic médical.
- Améliorer la gestion et la compréhension des données médicales multimodales.

### **Objectifs spécifiques :**

- Téléconsultation vidéo patient ↔ médecin.
- Chatbot médical intelligent basé sur RAG.
- Automatisation de la lecture et indexation de documents médicaux via OCR et embeddings vectoriels.
- Tableaux de bord pour suivi patient et praticien.
- Sécurité et conformité HIPAA/RGPD.

## ***4. Description fonctionnelle***

Fonctionnalités principales :

- Téléconsultation vidéo (WebRTC)
- Gestion des profils utilisateurs (patient, médecin, admin)
- Upload et OCR de documents médicaux
- Analyse multimodale : OCR, résumé de texte, recherche vectorielle
- Chatbot médical RAG
- Tableaux de bord interactifs
- Notifications et rappels automatisés

## ***5. Contraintes techniques***

**Architecture** : microservices

**Backend** : FastAPI (Python, async)

**Base de données** : PostgreSQL + pgvector

**Stockage** : MinIO (S3 compatible)

**Frontend** : React (Vite, CSS corporate)

**IA** : OpenAI / Ollama / Mixtral / PyMuPDF4LLM / Tesseract

**Sécurité** : JWT, TLS, RBAC, chiffrement

**Déploiement** : Docker Compose, Kubernetes, Cloud (AWS/Azure/GCP)

**CI/CD** : GitHub Actions

**Observabilité** : Grafana, Loki, OpenTelemetry

## **6. Utilisateurs ciblés**

- **Médecins praticiens** : diagnostic assisté et suivi clinique
- **Patients** : consultation à distance et gestion de leurs données
- **Administrateurs** : supervision, sécurité et gestion des accès

## **7. Livrables attendus**

- Application web complète et déployée
- Documentation technique (schémas BD, API REST, architecture)
- Manuel d'utilisation
- Présentation finale et démo
- Contrat de service web transactionnel

## **8. Calendrier prévisionnel**

**Semaine 1-2** : Analyse du besoin et conception UML

**Semaine 3-4** : Développement microservices (auth, patient, médecin)

**Semaine 5-6** : Intégration OCR et IA RAG

**Semaine 7-8** : Téléconsultation et tableaux de bord

**Semaine 9-10** : Tests, sécurité, CI/CD

**Semaine 11** : Démo finale et documentation

## **9. Critères de réussite**

- Plateforme stable et sécurisée
- Analyse multimodale fiable
- Réponses IA RAG cohérentes
- Interface fluide et ergonomique
- Conformité aux normes RGPD/HIPAA
- Livraison complète dans les délais