

Concevoir une fonctionnalité IA avec RAG, ontologies et orchestration

Référence: DEV-LLM-201

Tarif:

1400 € Net de taxes par bénéficiaire

Durée :

2 jours (14h) en présentiel / distanciel

Effectif:

4 à 8 participants

Niveau et public:

La formation et la certification s'adressent aux développeurs Backend, développeurs IA et développeurs FullStack expérimentés, ainsi qu'aux Lead Developers techniques souhaitant intégrer l'Intelligence Artificielle dans leurs projets, afin de créer des solutions innovantes performantes.

Pré-requis :

Avoir suivi le programme "Intégrer un LLM dans une application métier"

ou maîtriser l'intégration API LLM

Maîtriser un langage de programmation (Python recommandé)

Comprendre les API REST

Avoir des notions de base en prompt engineering

Modalités :

Formation en présentiel ou à distance, théorie et travaux pratiques.

Délais d'accès:

Inscription jusqu'à 7 jours avant la session (selon la disponibilité).

Accessibilité:

Adaptation possible pour les personnes en situation de handicap.

Contact:

iazen@iazenconsulting.onmicrosoft.com
+33 6 23 37 13 91

Objectif général:

- Permettre aux développeurs de concevoir, implémenter et sécuriser une fonctionnalité d'intelligence artificielle générative intégrant un pipeline RAG, des embeddings, une structuration ontologique et une orchestration via LangChain.

Objectifs pédagogiques:

À l'issue de la formation, le participant sera capable de :

- Concevoir un pipeline RAG complet et structuré
- Implémenter un système d'embeddings et une base vectorielle
- Structurer une ontologie métier adaptée à un cas d'usage
- Intégrer un graphe de connaissances dans un système LLM
- Orchestrer un workflow LLM avec LangChain
- Mettre en place une mémoire conversationnelle persistante
- Évaluer la robustesse et la qualité des réponses générées
- Optimiser les performances et les coûts d'un système RAG

Compétences visées :

- Concevoir une architecture RAG industrielle
- Implémenter un système de recherche vectorielle
- Structurer la connaissance métier via ontologies
- Orchestrer des chaînes et agents LLM
- Tester et sécuriser un système IA
- Analyser la cohérence et la pertinence des réponses

Programme détaillé:

Jour 1 – Structurer la donnée et concevoir un pipeline RAG

Concevoir une architecture RAG complète

- Expliquer les limites d'un LLM sans contexte externe
- Définir les principes du Retrieval-Augmented Generation
- Structurer un pipeline ingestion → chunking → embedding → indexation → génération
- Identifier les points critiques d'un pipeline RAG
- Choisir une stratégie de découpage documentaire adaptée

Implémenter un système d'embeddings

- Générer des embeddings à partir de documents métier Forcer un format de sortie structuré
- Comprendre la similarité vectorielle

Cas pratique : Rédaction d'un plan d'intégration et d'une feuille de route tenant compte des contraintes organisationnelles et techniques de l'entreprise.

Concevoir une fonctionnalité IA avec RAG, ontologies et orchestration

Formateur :

Amel Mhamdi est Data Scientist Senior et Architecte en IA, spécialisée dans les systèmes multi-agents et les modèles de langage (LLM). Elle possède plus de 10 ans d'expérience dans la conception de solutions d'intelligence artificielle, les architectures distribuées et le déploiement de projets data à fort impact. Experte en LangChain, en RAG, en IA générative et en orchestration d'agents, elle accompagne également des missions de conseil et de formation.

Modalités d'évaluation :

Évaluation continue via ateliers pratiques
Validation du projet final
Analyse collective des architectures proposées
Une attestation de fin de formation mentionnant les compétences acquises est remise au participant ayant suivi l'intégralité de la formation.

Processus de candidature :

Analyse du dossier de candidature, entretien de positionnement et validation des prérequis.

Moyens pédagogiques et techniques :

Supports pédagogiques, démonstrations, cas pratiques et environnement de développement (Python, LLM, LangChain).

Indicateurs de résultat :

Taux de satisfaction et de réussite, avec évaluation des compétences acquises en fin de formation.

- Sélectionner une base vectorielle adaptée (FAISS, Chroma, Pinecone)
- Indexer des documents structurés et non
- Interroger un index vectoriel

Structurer la connaissance via ontologies et graphes

- Définir des entités métier pertinentes
- Identifier les relations entre concepts
- Expliquer la différence entre base documentaire et graphe de connaissances
- Intégrer un graphe dans un pipeline RAG (GraphRAG conceptuel)

Atelier pratique – Concevoir un assistant documentaire structuré

- Charger et découper un corpus métier
- Générer des embeddings
- Indexer dans une base vectorielle
- Interroger le système avec contexte enrichi
- Intégrer une structuration ontologique simple

Jour 2 – Orchestrer, sécuriser et évaluer un système IA

Orchestrer un système LLM avec LangChain

- Comprendre les composants principaux de LangChain
- Configurer une chaîne LLM (LLMChain)
- Définir des PromptTemplate structurés
- Ajouter une mémoire conversationnelle
- Intégrer des tools externes

Mettre en place une orchestration avancée

- Structurer un workflow multi-étapes
- Implémenter une logique conditionnelle
- Produire des sorties structurées complexes
- Gérer les exceptions et cas limites

Évaluer la robustesse du système

- Identifier les risques d'hallucination
- Tester des scénarios adversariaux
- Comparer des réponses générées
- Mesurer la cohérence et la pertinence
- Mettre en place une logique de scoring simple

Concevoir une fonctionnalité IA avec RAG, ontologies et orchestration

Optimiser les performances et les coûts

- Réduire la taille des contextes injectés
- Mettre en place des stratégies de cache
- Analyser la consommation de tokens
- Identifier les facteurs d'optimisation

Projet final – Concevoir une fonctionnalité IA complète

Concevoir une fonctionnalité métier intégrant :

- Un moteur RAG vectoriel
- Une structuration ontologique
- Une orchestration LangChain
- Une sortie structurée validée
- Une logique de gestion des erreurs

Présenter l'architecture conçue et justifier les choix techniques.

Méthodes pédagogiques

- Alternance théorie / pratique (40 % / 60 %)
- Démonstrations techniques encadrées
- Ateliers progressifs
- Travail en petits groupes
- Revue technique collective

Livrables remis aux participants

- Supports de formation détaillés
- Exemples de pipelines RAG
- Modèles d'ontologies simplifiées
- Templates LangChain
- Scripts de démonstration
- Checklist d'évaluation de robustesse
- Guide d'optimisation des coûts LLM