Présentiel/distanciel

NOUVEAUTÉ

IA générative pour développeurs Intégrer des LLM en production avec API, agents, RAG et multi-fournisseurs

Objectifs pédagogiques

- Identifier les principes techniques des modèles de langage et leurs impacts sur une application
- Analyser les différences entre les principaux fournisseurs de modèles LLM et leurs API
- Construire des prompts structurés, transférables et exploitables dans un contexte applicatif
- Utiliser les SDK Python et TypeScript pour intégrer des API LLM dans une application
- Implémenter des sorties structurées, du streaming, du batch, du cache et une gestion robuste des erreurs
- Concevoir une abstraction multi-fournisseurs pour limiter le verrou technologique
- Développer des agents utilisant des outils, du function calling et des workflows contextualisés
- Mettre en oeuvre une architecture RAG adaptée à un cas d'usage documentaire
- Configurer un environnement de développement assisté par IA avec Claude Code ou des outils équivalents
- Appliquer des pratiques de sécurisation pour les applications intégrant de l'IA générative
- Déployer un prototype fonctionnel intégrant LLM, agents, RAG et logique applicative

Prérequis

Les participants doivent disposer des connaissances suivantes :
- Maîtriser un langage de programmation, idéalement Python ou JavaScript / TypeScript
- Utiliser Git et de la ligne de commande
- Comprendre les principes des API HTTP / REST
- Manipuler des données au format JSON
- Avoir déjà intégré ou consommé une API applicative

Une première expérience avec des outils d'IA générative ou des assistants de développement est un plus, mais n'est pas obligatoire.

Public concerné

Cette formation s'adresse aux profils techniques impliqués dans le développement d'applications intégrant de l'intelligence artificielle générative :
- Développeurs back-end
- Développeurs front-end
- Développeurs full-stack
- Développeurs mobile
- Software engineers
- Tech leads et architectes applicatifs
- Développeurs amenés à intégrer des modèles de langage dans des services, produits numériques ou applications métiers

Bénéfices pour les participants : passer d'un usage ponctuel de l'IA générative à une approche structurée, industrialisable et orientée production, en intégrant des LLM dans des applications réelles, avec une attention portée aux performances, à la sécurité, à la maintenabilité et au choix des fournisseurs.

Programme détaillé

Panorama IA générative et fondamentaux techniques (7h00)

- Comprendre le fonctionnement général des modèles de langage
  - Rôle des modèles de langage dans les applications modernes
  - Différences entre IA générative, assistants conversationnels et intégration applicative
  - Principes généraux d'un LLM : entrée, contexte, génération, sortie
  - Cas d'usage développeur : génération de contenu, analyse de texte, extraction d'information, automatisation, assistance au développement
- Identifier les fondamentaux d'architecture des LLM
  - Architecture de type transformer
  - Notion de tokens et impacts sur les coûts, les limites de contexte et la performance
  - Fenêtre de contexte, mémoire conversationnelle et limites pratiques
  - Température, top-p et paramètres influençant la génération
  - Latence, débit, qualité de réponse et arbitrages techniques
- Comparer les principaux modèles et fournisseurs
  - Panorama des modèles disponibles chez OpenAI, Anthropic, Google et Mistral
  - Différences entre modèles généralistes, modèles rapides, modèles multimodaux et modèles orientés raisonnement
  - Critères de choix : performance, coût, latence, disponibilité, confidentialité, fonctionnalités API
  - Risques liés au verrou fournisseur et intérêt d'une approche multi-modèles
- Évaluer les coûts, performances et contraintes d'intégration
  - Compréhension des modèles de tarification par tokens
  - Estimation du coût d'un cas d'usage applicatif
  - Optimisation des prompts pour réduire la consommation inutile
  - Bonnes pratiques pour mesurer latence, qualité et stabilité des réponses
- Travail pratique : cartographier un cas d'usage LLM intégrable dans une application
  - Les participants analysent un besoin applicatif réaliste, par exemple un assistant de support, un analyseur de documents ou un générateur de synthèses métiers. Ils identifient les entrées, les sorties attendues, les contraintes de qualité, les risques, les coûts estimés, le modèle le plus adapté et les indicateurs de succès à suivre. Le livrable attendu est une fiche de cadrage technique permettant de préparer l'intégration du cas d'usage dans une application.

Prompting structuré et transférable pour applications LLM (7h00)

- Construire des prompts robustes et réutilisables
  - Différences entre prompt utilisateur, prompt système et messages de contexte
  - Structuration d'un prompt pour une application métier
  - Rôle des consignes, exemples, contraintes et formats de sortie
  - Techniques de clarification, de cadrage et de réduction de l'ambiguïté
- Concevoir des system prompts adaptés à la production
  - Définir un rôle, un périmètre fonctionnel et des règles de comportement
  - Encadrer les réponses selon les besoins métier
  - Prévenir les sorties hors sujet, incohérentes ou non exploitables
  - Tester la stabilité d'un prompt sur plusieurs cas d'entrée
- Produire des sorties structurées exploitables par le code
  - JSON, schémas de données et formats contrôlés
  - Différences entre texte libre et structured outputs
  - Contraintes de validation côté application
  - Gestion des réponses incomplètes, invalides ou ambiguës
- Adapter le prompting aux différents modèles
  - Comprendre le déterminisme relatif selon les fournisseurs
  - Adapter une consigne à plusieurs modèles sans perdre la logique fonctionnelle
  - Comparer les résultats obtenus sur un même scénario
  - Identifier les écarts de comportement et leurs impacts applicatifs
- Travail pratique : concevoir une bibliothèque de prompts transférables
  - Les participants conçoivent un ensemble de prompts pour un même cas d'usage : classification, extraction structurée, synthèse et génération de réponse. Ils testent ces prompts sur plusieurs modèles, comparent les résultats, ajustent les consignes et formalisent une version réutilisable dans une application. Le livrable attendu est une mini-bibliothèque de prompts documentée, accompagnée de critères de validation et d'exemples d'entrées / sorties.

API LLM multi-fournisseurs, SDK et intégration applicative (7h00)

- Utiliser les principales API LLM du marché
  - Principes d'appel API pour les modèles de langage
  - Anthropic Messages API
  - OpenAI Responses API
  - Google Gemini API
  - Mistral API
  - Points communs et différences dans les formats de requêtes, les réponses et les paramètres
- Intégrer les SDK Python et TypeScript
  - Installation et configuration des SDK
  - Gestion des clés API et variables d'environnement
  - Construction d'un client minimal

- Structuration du code pour isoler les appels aux fournisseurs
- Implémenter les fonctionnalités avancées d'intégration
  - Streaming des réponses pour améliorer l'expérience utilisateur
  - Batch et traitements asynchrones
  - Vision et analyse d'images selon les modèles disponibles
  - Traitement documentaire et extraction d'informations
  - PDF, texte long et contraintes de découpage du contexte
- Fiabiliser les appels aux modèles
  - Prompt caching et optimisation des appels récurrents
  - Gestion des erreurs, timeouts et limites de débit
  - Stratégies de retry et fallback
  - Journalisation des appels et traçabilité technique
  - Tests automatisés sur les sorties structurées
- Concevoir une abstraction multi-LLM
  - Intérêt d'une couche d'abstraction entre l'application et les fournisseurs
  - Présentation des approches de type LiteLLM, OpenRouter et Vercel AI SDK
  - Normalisation des entrées et sorties
  - Choix d'une stratégie multi-fournisseurs selon les besoins métier
  - Intégration progressive dans une application existante
- Travail pratique : intégrer plusieurs fournisseurs LLM dans une application existante
  - Les participants partent d'une application simple existante et ajoutent une couche d'appel LLM compatible avec plusieurs fournisseurs. Ils implémentent au minimum deux connecteurs, une sortie structurée, un mode streaming ou batch, une gestion des erreurs et un mécanisme de fallback. Le livrable attendu est une intégration fonctionnelle permettant de basculer d'un fournisseur à l'autre sans modifier la logique métier principale.

Agents, tool use, MCP et architectures RAG (7h00)

- Comprendre les principes des agents applicatifs
  - Différence entre chatbot, workflow automatisé et agent
  - Boucle de raisonnement, action, observation et réponse
  - Cas d'usage pertinents pour les agents
  - Limites opérationnelles et risques de sur-automatisation
- Mettre en oeuvre le function calling et le tool use
  - Définition d'outils appelables par un modèle
  - Implémentations avec Anthropic, OpenAI et Mistral
  - Schémas d'entrée, validation des paramètres et contrôle côté application
  - Sécurisation des actions déclenchées par un modèle
  - Gestion des erreurs et confirmations utilisateur
- Exploiter le Model Context Protocol
  - Rôle du MCP dans la connexion entre modèles et outils
  - Consommation de serveurs MCP existants
  - Conception d'un serveur MCP custom
  - Réutilisation d'un serveur MCP entre plusieurs modèles ou environnements
  - Contraintes de sécurité, permissions et isolation
- Concevoir une architecture RAG opérationnelle
  - Principes du Retrieval-Augmented Generation
  - Différence entre contexte fourni, recherche documentaire et mémoire applicative
  - Critères de choix d'une architecture RAG
  - Découpage documentaire, indexation, recherche sémantique et réinjection du contexte
  - Gestion du long context : bénéfices, limites et arbitrages
- Comparer RAG, long context et agents outillés
  - Quand utiliser un RAG plutôt qu'un long contexte
  - Quand ajouter un agent plutôt qu'un simple appel modèle
  - Combiner retrieval, outils et raisonnement contrôlé
  - Mesurer la pertinence des réponses et limiter les hallucinations
  - Structurer un routage entre modèles, outils et sources documentaires
- Travail pratique : développer un agent multi-modèles avec routage et RAG
  - Les participants conçoivent un agent capable de répondre à une demande utilisateur en choisissant entre plusieurs actions : appeler un modèle, interroger une base documentaire, utiliser un outil métier ou demander une clarification. Ils mettent en place une logique de routage, une source documentaire exploitable en RAG, un outil applicatif simple et des garde-fous de validation. Le livrable attendu est un agent fonctionnel capable de traiter un scénario métier contextualisé avec justification des choix techniques.

Claude Code, développement assisté par IA et mise en production (7h00)

- Intégrer Claude Code et les outils équivalents dans le workflow développeur
  - Présentation de Claude Code en ligne de commande
  - Positionnement par rapport à Cursor, GitHub Copilot, Continue et Aider
  - Cas d'usage : compréhension de code, génération, refactoring, tests, documentation, revue
  - Bonnes pratiques pour conserver la maîtrise du code produit
  - Limites des assistants de développement et risques associés
- Configurer un environnement de développement assisté par IA
  - Connexion au dépôt de code et contexte projet
  - Paramétrage des outils, permissions et accès
  - Utilisation du MCP dans un environnement de développement
  - Gestion des hooks, sous-agents et compétences spécialisées
  - Structuration des tâches pour améliorer la qualité des résultats
- Sécuriser les applications intégrant l'IA générative

- Risques de prompt injection
- Gestion des secrets, clés API et variables sensibles
- Validation des entrées et sorties du modèle
- Sandboxing et limitation des actions dangereuses
- Traçabilité, supervision et journalisation
- Revue humaine et contrôle des décisions critiques

• Préparer l'industrialisation d'un prototype IA
- Passage d'un prototype à une intégration maintenable
- Organisation du code et séparation des responsabilités
- Tests fonctionnels et tests de non-régression sur les prompts
- Supervision de la latence, des coûts et des erreurs
- Documentation technique et transmission aux équipes

• Travail pratique : réaliser un POC déployé end-to-end
- Les participants finalisent un prototype complet intégrant les composants clés de la formation : appel LLM, prompts structurés, couche multi-fournisseurs, agent outillé, RAG, validation des sorties et mesures de sécurité de base. Ils utilisent un assistant de développement IA pour accélérer certaines tâches, tout en contrôlant les modifications produites et en documentant les choix réalisés. Le livrable attendu est un POC fonctionnel, démontrable et structuré, accompagné d'une note technique présentant l'architecture, les risques identifiés, les choix fournisseurs et les pistes d'industrialisation.

Délais d'accès à la formation

Les inscriptions sont possibles jusqu'à 48 heures avant le début de la formation.
Dans le cas d'une formation financée par le CPF, ENI Service est tenu de respecter un délai minimum obligatoire de 11 jours ouvrés entre la date d'envoi de sa proposition et la date de début de la formation.

Modalités et moyens pédagogiques, techniques et d'encadrement

Formation avec un formateur, qui peut être suivie selon l'une des 3 modalités ci-dessous :
1 Dans la salle de cours en présence du formateur.
2 Dans l'une de nos salles de cours immersives, avec le formateur présent physiquement à distance. Les salles immersives sont équipées d'un système de visio-conférence HD et complétées par des outils pédagogiques qui garantissent le même niveau de qualité.
3 Depuis votre domicile ou votre entreprise. Vous rejoignez un environnement de formation en ligne, à l'aide de votre ordinateur, tout en étant éloigné physiquement du formateur et des autres participants. Vous êtes en totale immersion avec le groupe et participez à la formation dans les mêmes conditions que le présentiel. Pour plus d'informations : Le téléprésentiel notre solution de formation à distance.

Le nombre de stagiaires peut varier de 1 à 12 personnes (5 à 6 personnes en moyenne), ce qui facilite le suivi permanent et la proximité avec chaque stagiaire.
Chaque stagiaire dispose d'un poste de travail adapté aux besoins de la formation, d'un support de cours et/ou un manuel de référence au format numérique ou papier.
Pour une meilleure assimilation, le formateur alterne tout au long de la journée les exposés théoriques, les démonstrations et la mise en pratique au travers d'exercices et de cas concrets réalisés seul ou en groupe.

Modalités d'évaluation des acquis

En début et en fin de formation, les stagiaires réalisent une auto-évaluation de leurs connaissances et compétences en lien avec les objectifs de la formation. L'écart entre les deux évaluations permet ainsi de mesurer leurs acquis.
En complément, le formateur évalue chaque stagiaire sur l'atteinte des objectifs pédagogiques de la formation selon quatre niveaux (non évalué, non acquis, en cours d'acquisition, acquis). Cette évaluation repose sur une modalité choisie par le formateur en cohérence avec la formation : QCM, exercices pratiques réalisés pendant la formation, évaluation finale de synthèse, quiz interactif de validation, étude de cas, mise en situation, analyse de l'auto-évaluation, autres modalités adaptées.
Pour les stagiaires qui le souhaitent, certaines formations peuvent être validées officiellement par un examen de certification. Les candidats à la certification doivent produire un travail personnel important en vue de se présenter au passage de l'examen, le seul suivi de la formation ne constitue pas un élément suffisant pour garantir un bon résultat et/ou l'obtention de la certification.
Pour certaines formations certifiantes (ex : ITIL, DPO, ...), le passage de l'examen de certification est inclus et réalisé en fin de formation. Les candidats sont alors préparés par le formateur au passage de l'examen tout au long de la formation.

Moyens de suivi d'exécution et appréciation des résultats

Feuille de présence, émargée par demi-journée par chaque stagiaire et le formateur.
Evaluation qualitative de fin de formation, qui est ensuite analysée par l'équipe pédagogique ENI.
Attestation de fin de formation, remise au stagiaire en main propre ou par courrier électronique.

Qualification du formateur

La formation est animée par un professionnel de l'informatique et de la pédagogie, dont les compétences techniques, professionnelles et pédagogiques ont été validées par des certifications et/ou testées et approuvées par les éditeurs et/ou notre équipe pédagogique.
Il est en veille technologique permanente et possède plusieurs années d'expérience sur les produits, technologies et méthodes enseignés.
Il est présent auprès des stagiaires pendant toute la durée de la formation.

Accessibilité de la formation