

# Curriculum

## Expert en Développement d'Applications Full Stack

### C# (.NET 8) | Angular | Python (FastAPI) | IA & Cloud

Ce parcours est conçu pour transformer un débutant en un ingénieur logiciel capable de concevoir, déployer et maintenir des architectures distribuées modernes, en exploitant la puissance de l'Intelligence Artificielle.

#### PHASE 1 : Fondations de l'Ingénierie logicielle

*L'objectif est de comprendre le "comment" avant le "quoi".*

##### Module 01 : Architecture des systèmes et Bas niveau

- **Architecture Matérielle** : Fonctionnement interne (CPU, registres, hiérarchie de mémoire RAM/Cache).
- **Codage de l'Information** : Encodages (Binaire, Hex, UTF-8) et gestion des types de données en mémoire.
- **Systèmes d'Exploitation** : Gestion des processus, file systems et ordonnancement.
- **Runtime .NET** : Comprendre la compilation MSIL (Intermediate Language) et le moteur JIT (Just-In-Time).

##### Module 02 : Algorithmique avancée et Structures de données

- **Logique de Programmation** : Algèbre de Boole, structures conditionnelles et itératives complexes.
- **Collections de Données** : Listes, Dictionnaires, Piles (Stacks), Files (Queues) et Arbres.
- **Analyse de Performance** : Introduction à la notation Big O et optimisation de la complexité.

**Module 03 : Programmation Orientée Objet (POO) avec C# 12**

- **Concepts Fondamentaux** : Classes, Objets, Membres et Encapsulation stricte.
- **Piliers de l'Objet** : Héritage, Polymorphisme et Abstraction.
- **Design par Contrat** : Interfaces et classes abstraites pour le découpage modulaire.
- **Robustesse** : Stratégies de gestion d'exceptions et logging.

**Module 04 : Ingénierie des données relationnelles**

- **Modélisation Conceptuelle** : Diagrammes de classes (UML) et schémas Entité-Relation.
- **Normalisation** : Conception de bases de données performantes (1NF, 2NF, 3NF).
- **Maîtrise du SQL** : DDL (Définition), DML (Manipulation) et requêtes complexes (Jointures, Sous-requêtes).
- **Objets de Base de Données** : Vues, Indexation et Procédures Stockées (T-SQL).

## ● PHASE 2 : Écosystème professionnel et Applications Desktop

*Passage de la console à l'interface graphique et aux outils collaboratifs.*

### Module 05 : Workflow collaboratif et DevOps Local

- **Maîtrise de l'IDE** : Configuration avancée de Visual Studio et techniques de débogage.
- **Gestion de Versions** : Workflow Git professionnel (Branches, Merging, Rebase).
- **Collaboration Distribuée** : Pull Requests et Code Reviews sur GitHub/Azure DevOps.
- **Gestion de Dépendances** : NuGet pour .NET et Pip pour Python.

### Module 06 : Projet 1 - Système de gestion en Console (C# / LINQ)

- **Architecture N-Tier** : Séparation claire entre Logique Métier (BLL) et Accès Données (DAL).
- **Maîtrise de LINQ** : Requêtage d'objets et manipulation de flux de données.
- **Persistence** : Sérialisation complexe en JSON et XML.

### Module 07 : Projet 2 - Client lourd avec Windows Forms & SQLite

- **UX Desktop** : Contrôles avancés (DataGridView), gestion des layouts dynamiques.
- **Programmation Événementielle** : Gestion du cycle de vie des formulaires et délégués.
- **Reporting** : Génération de documents et exports de données.
- **Persistence Locale** : Implémentation d'un CRUD complet sur SQLite.

## ● PHASE 3 : Architecture Web et Productivité IA

*Entrée dans le monde du Web moderne et de l'assistance par IA.*

### **Module 08 : L'IA au service du développement (Dev Productivity)**

- **Prompt Engineering** : Techniques de génération de code, refactoring et documentation.
- **Pair Programming IA** : Utilisation intensive de GitHub Copilot et ChatGPT.
- **Validation Critique** : Audit de sécurité et détection d'hallucinations dans le code généré.

### **Module 09 : Projet 3 - Application Web moderne avec ASP.NET Core & EF Core**

- **Architecture Web API** : Routage REST, Contrôleurs et Injection de Dépendances.
- **Entity Framework Core** : Mapping Objet-Relationnel (ORM), Migrations et Lazy/Eager Loading.
- **Sécurité Applicative** : JWT (JSON Web Tokens), Authentification et Autorisation.
- **Middleware & Filtres** : Personnalisation du pipeline de requête.

## ● PHASE 4 : Architectures Scalables (Angular, Python & Cloud)

*Expertise Full Stack et conteneurisation.*

### Module 10 : Cloud, Conteneurs et SaaS

- **Conteneurisation** : Dockerisation d'applications et Docker Compose.
- **Cloud Computing** : Concepts IaaS/PaaS/SaaS avec focus sur Microsoft Azure.
- **Architecture SaaS** : Design multi-tenants et isolation de données.
- **CI/CD** : Pipelines de déploiement automatisé avec GitHub Actions.

### Module 11 : Qualité logicielle et Principes SOLID

- **Clean Architecture** : Application des principes SOLID et Design Patterns.
- **Tests Unitaires & Mocking** : Frameworks xUnit et Moq pour l'isolation.
- **TDD (Test Driven Development)** : Introduction à la conception pilotée par les tests.

### Module 12 : L'Écosystème Python moderne

- **Syntaxe Avancée** : List Comprehensions, Decorators et gestion asynchrone.
- **Data Engineering** : Introduction à Pandas pour l'analyse de données.
- **Backend Python** : Création d'APIs ultra-rapides avec FastAPI.

### Module 13 : Développement frontend avec Angular (v17+)

- **Composants & Templates** : Data Binding, Directives et Pipes.
- **Programmation Réactive** : Maîtrise de RxJS et des Observables.
- **Architecture d'État** : Services, Signals et Injection de Dépendances.
- **Navigation** : Routage avancé, Lazy Loading et Guards de sécurité.

### Module 14 : Projet 4 - Application hybride (Python API / Angular)

- **Architecture Découplée** : Backend FastAPI sécurisé et Frontend Angular dynamique.
- **Documentation** : Spécification OpenAPI (Swagger) automatique.
- **Qualité** : Tests d'intégration et validation des flux de données.

## **PHASE 5 : Expertise finale**

*La consécration du parcours.*

### **Module 15 : Projet final de synthèse (Portfolio de sortie)**

- **Conception** : Rédaction d'un dossier d'architecture technique complet.
- **Développement** : Création d'une solution Full Stack (C# ou Python au choix) avec Frontend Angular.
- **Déploiement** : Mise en ligne sur le Cloud.