

Concepteur·rice Développeur·euse d'Applications

Préparation au Titre Professionnel :
« TP - Concepteur Développeur d'Applications »

Reconnu par l'État de niveau 6 (ou niveau BAC+3)
Inscrit au RNCP (arrêté du 26/04/2023, J.O. du 13/05/2023)

Code RNCP : RNCP37873 – Code NSF : 326t – Formacode : 31067, 31036, 31010, 31084, 31006

Objectifs de formation

Prendre en charge les études fonctionnelles et techniques,
Spécifier et concevoir techniquement une application informatique
de type client/serveur, n-tiers ou mobile,
Programmer une application informatique,
Participer à la gestion de projet informatique.

Public et prérequis

- Jeune diplômé en informatique Bac +2
- Informaticien expérimenté

Modalités et délais d'accès

- **Formation continue** : plusieurs rentrées par an. N'hésitez pas à nous consulter.
- **Alternance** (contrat de professionnalisation ou d'apprentissage) : rentrées tout au long de l'année.
- **Compte personnel de formation** (CPF).

Il vous faudra assister à une réunion d'information qui sera suivie d'un entretien avec le service relations École / Entreprises ainsi que des tests de logique et de positionnement.

Durée de la formation

En continu :

- Formation en centre : 1120 heures
- Stage pratique en entreprise : 280 heures
- Durée totale de la formation : 1400 heures

En alternance :

- Formation en centre : 490 heures
- Sur un contrat d'environ 15 mois

Tarifs

Les tarifs dépendent de la modalité d'accès.
N'hésitez pas à nous consulter

Contact

ecole@eni-ecole.fr

N.B. : Dans un souci de lisibilité, le masculin utilisé dans ces différents textes pourra également désigner le féminin, et ce, sans recours systématique à l'écriture inclusive.

Moyens pédagogiques - techniques - d'encadrement

Équipements pédagogiques :

- Un poste de travail par apprenant équipé (en quasi-totalité) :
 - d'un processeur i5
 - d'un disque NVMe (32 Go de RAM)
- Un tableau blanc interactif installé dans chaque salle de formation
- Une photocopieuse/imprimante en libre accès
- Un accès internet sur chaque poste de travail
- Un serveur et un commutateur Ethernet Gigabit dans chaque salle
- Dix routeurs Cisco

Moyens pédagogiques :

- Travaux dirigés après chaque phase de cours : explications et démonstrations par le formateur et exécutés ensuite par les stagiaires.
- Travaux pratiques pour que le stagiaire apprenne à appliquer seul ce qu'il a appris et cherche par lui-même.
Mises en situation professionnelle avec des cas spécifiques « entreprise ».
- Création de plateaux techniques qui recréent les conditions d'activités réelles des entreprises.
- Études de cas où le stagiaire doit résoudre une problématique technique ou managériale.
- Exposés oraux où le stagiaire doit préparer une présentation sur un thème technique donné.

Supports pédagogiques :

- Supports ENI Editions/internes pour chaque cours
- Un accès la Bibliothèque Numérique des Editions
- Un accès aux agréments techniques des éditeurs (Microsoft IT Academy et Microsoft Imagine Premium, Cisco Academy)
- Un accès aux revues informatiques
- Un accès au programme Microsoft Azure (licences logiciels Microsoft)
- Un accès à la plateforme [goFluent](#) (plateforme d'autoformation et programme individualisé en anglais)
- Systèmes d'exploitation Microsoft, GNU/Linux et Cisco IOS
- Logiciels bureautiques (Microsoft Office 365 : Word, Excel, Powerpoint, ...)
- Outils de développement (Visual Studio, Oracle, Eclipse, Hibernate, Tomcat, Java...)
- Systèmes de gestion de bases de données relationnelles (MySQL, SQL Server, ...)

Suivi et évaluation :

Les blocs de compétences constituant le titre « TP - Concepteur Développeur d'Applications » sont les suivants :

- Développer la partie *front-end* d'une application web ou web mobile en intégrant les recommandations de sécurité
- Développer la partie *back-end* d'une application web ou web mobile en intégrant les recommandations de sécurité
- Préparer le déploiement d'une application en intégrant les recommandations de sécurisée

Évaluations en cours de formation mesurant compétence par compétence le degré de maîtrise de la situation professionnelle concernée : devoirs écrits, mises en situation professionnelle, QCM, études de cas, exposés oraux...

Épreuve finale : rédaction d'un rapport d'activité – soutenance orale devant un jury de professionnels extérieurs à l'École.

Possibilité d'accès au titre par capitalisation de blocs de compétences.

Débouchés et métiers

Le **Concepteur Développeur d'Applications** anime et conduit la réalisation de projets :

- en prenant en charge les études fonctionnelles et techniques,
- en participant à la réalisation et à la maintenance des applications informatiques,
- en participant à la gestion de ces projets.

Métiers :

- Analyste Concepteur
- Concepteur Développeur Web
- Concepteur Développeur d'Applications mobiles
- ...

Équivalences et passerelles

La formation Concepteur·rice Développeur·euse d'Applications prépare au titre RNCP du même nom :

<https://www.francecompetences.fr/recherche/rncp/37873/>

Vous pouvez également poursuivre votre cursus de formation au sein de notre établissement (lien vers tableau des formations).

Programme résumé

• Liste des modules de cours :

- Le langage de requête SQL avec SQL Server
- Le développement Web Côté Serveur – Back-End avec Java Spring Boot
- Le SQL avancé avec Transact SQL et Sécurité
- La Gestion de projet et les Techniques de Communication
- Les Frameworks de Java : Spring Security et ORM
- Utilisation avancée du Framework JS avec Angular
- Analyse et Conception avec Oracle Data Modeler
- Projet Web FullStack : Java Spring Boot avec Angular
- Découverte de la Technologie Cross-Plateform avec Flutter
- DevOps - Infrastructure et déploiement d'applications
- L'Intelligence Artificielle avec Python + Projet Final

• *Optionnel via Campus En Ligne :*

- La Programmation avec Python
- Le développement web côté serveur avec ASP.Net Core

Programme détaillé

Le langage de requête SQL avec SQL Server

- Introduction au SQL
- L'environnement
- La gestion des tables (DDL)
- La mise à jour des données (DML)
- L'extraction des données (DML)
- Les transactions

Introduction au développement Web côté Serveur avec Java Spring Boot

- Spring - Développer une application web classique organisée en couches
- Créer et configurer l'application
- Couplage fort/faible
- Structurer l'application en couche avec Spring Core • Spring Web
- Développer la couche IHM d'une application web
- Spring JDBC - Développer la Couche d'accès aux données
- Situer la couche BLL
- Sécuriser une application web (authentification et autorisations)

Le SQL avancé avec Transact SQL et Sécurité

- **Le SQL avancé :**
 - Introduction sur le langage procédural en SQL
 - Le Transact-SQL
 - Les procédures stockées
 - Les fonctions stockées
 - Les déclencheurs
 - Import et export de donnée (dump, CSV) (optionnel)
- **Sécurité Informatique :**
 - Introduction à la sécurité informatique
 - Attaque/Défense sur divers scénarios (Injections, accès de contrôle, ...)

La Gestion de projet et les Techniques de Communication

- Introduction à la gestion de projet
- Organisation et parties prenantes
- Méthodes de gestion de projet
- Le cahier des charges
- Découper un projet en tâches
- Les charges de travail
- Le démarrage d'un projet
- Le pilotage d'un projet
- Techniques de communication

- **Rappels et introduction sur Spring Boot et architecture globale REST :**
 - Bases et l'architecture des projets Spring Boot
 - Web services
 - API Rest
- **Spring Data JPA :**
 - Utilisation de l'ORM
 - Tests unitaires
 - Gestion des logs
- **Web services – REST controller :**
 - API complète avec Rest controller
 - Tests unitaires, gestion des logs
- **Spring Data Rest (API minimaliste) :**
 - Hateoas, API minimaliste, url -> datas
 - Tests unitaires
 - Gestion des logs
- **Spring Security :**
 - Sécurisation de l'API
 - JWT
 - OAuth2
 - Tests unitaires
 - Gestion des logs
- **Swagger (optionnel) :**
 - Génération de la documentation de l'API
- **Déploiement (optionnel)**
- **AOP (optionnel) :**
 - Programmation orientée aspect

Utilisation avancée du Framework JS avec Angular

- **Introduction :**
 - Rappels sur le TypeScript
 - Introduction sur les frameworks et bibliothèques front-end
- **Introduction à Angular :**
 - L'architecture
 - Les modules et les composants
- **Les composants :**
 - Création, cycle de vie, communication entre composants
- **L'affichage dynamique :**
 - Les directives
 - Les pipes
- **Les services :**
 - Création
 - Logique métier
 - Injection de dépendances
- **Les routes :**
 - Comprendre le routing et la navigation
 - Créer des routes simples et paramétrées
 - Route Guards
- **Les formulaires :**
 - Création de formulaires basiques
 - La validation de données
 - La validation asynchrone
- **Travailler avec une API REST :**
 - Rappels sur le protocole http et les méthodes (get, post, put, ...)
 - Module HttpClient
 - Gestion des erreurs
 - Réponses asynchrones
- **Les observables :**
 - Comprendre le pattern Observer
 - Mise en application (Définition des Observers, Souscription, gestion des erreurs)
- **RxJS :**
 - Comprendre les bases de RxJS
 - Les opérateurs (créations, combinatoires, filtres, transformations, ...)
 - Gestion des erreurs
- **Les bonnes pratiques (optionnel) :**
 - Les conventions de nommage (Angular, RxJS)
 - Amélioration de la structure par modules et lazy loading

Analyse et Conception avec Oracle Data Modeler

- Projet informatique
- Le projet informatique (le processus)
- UML (Unified Modeling Language)
- L'expression initiale du besoin
- L'analyse détaillée des exigences
- Conception
- L'analyse des données (du modèle conceptuel au modèle physique)
- Les différents types de test
- Mise en place tests unitaires (ex. : JUnit, Mockito, TDD, ...)

Projet Web FullStack : Java Spring Boot avec Angular

(Durée de 2 semaines)

Objectifs :

- Développer les différentes couches de l'application (interface utilisateur, logique métier, accès aux données, ...).
- Créer et peupler la base de données avec des jeux de données de test.
- Créer des livrables : package à déployer, documentation technique
- Organiser son temps, planifier les tâches.
- Travailler en équipe avec un outil de versioning (ex. : GIT, ...).
- Présentation et restitution du travail

Description :

- Pendant ces deux semaines, les étudiant.es travailleront sur un projet *fullstack*, où ils développeront une application web en utilisant Java Spring Boot pour la partie backend, et *Angular* pour le frontend consolidant ainsi leurs compétences en développement d'applications web modernes.
- Ce projet leur permettra d'appréhender la conception et l'implémentation de l'architecture RESTful, d'intégrer une interface utilisateur réactive, et de gérer la persistance des données à l'aide de Java avec les bases de données relationnelles.
- L'objectif est de renforcer leur maîtrise des *frameworks Java* et *Angular* tout en travaillant en équipe, en appliquant les principes de développement *Agile* et en mettant en œuvre un système de gestion de version tel que *Git* pour le suivi des modifications.

Découverte de la Technologie Cross-Platform avec Flutter

- **Introduction :**
 - Le développement mobile
 - La place de Flutter sur le marché
- **Installation :**
 - Android Studio
 - SDK Flutter
 - Paramétrage des variables d'environnement
- **Première application et architecture :**
 - Découverte de l'architecture (pubspec, dossiers Android et iOS)
 - Découverte des premiers widgets
 - Découverte de la gestion des états
- **Dart :**
 - Introduction rapide (origine du langage, ...)
 - Les types (Evoquer le Sound Null Safety)
 - Fonctions
 - POO (constructeur (new non obligatoire, params optionnels, constructeurs nommés, ...)
 - Généricité
 - Exception
 - Traitement asynchrone
- **Les widgets :**
 - Introduction rapide (Material et Cupertino)
 - Widgets de placements et de mise en forme (Scaffold, Padding, Container, scrollview, Column, ...)
 - Widgets de contenus (Texte, boutons, images, ...)
- **Formulaire :**
 - Les éléments du formulaire (Champs de saisie, Switch, Radio, Checkbox)
 - Validation de données (GlobalKey, TextFormField, validator)
- **Le routing :**
 - Introduction (principe de la pile)
 - Routes Les états (locaux) :
 - Le widget StatefulWidget
 - Le cycle de vie
 - Le setState
- **Web services (REST) :**
 - Package http
- **Gestion avancée des états (optionnel) :**
 - Architecture MVVM (et/ou VIPER) et problématique des états locaux.
Difficulté de gestion des contextes et arbres des widgets
- **L'utilisation d'API (optionnel) :**
 - Appareil phot
 - GPS

DevOps - Infrastructure et déploiement d'applications

- Introduction au DevOps
- Base de Linux
- Virtualisation
- Conteneurisation avec Docker

L'Intelligence Artificielle avec Python + Projet Final

(Durée de 2 semaines)

- **Introduction au Machine Learning :**
 - Présentation
 - L'environnement technique
 - Les bibliothèques Python pour le ML (NumPy, Pandas, Scikit-Learn, ...)
 - Présentation des Datasets
 - Le travail sur les données
 - L'apprentissage supervisé
 - Les algorithmes de classification
 - L'apprentissage non-supervisé
 - Concepts avancés sur l'apprentissage
 - **Bilan :** Les risques et les limitations de l'IA, les enjeux éthiques
- **Projet Final (TP Synthèse) :** Petite compétition en mode *Kaggle*.

Option facultative

La Programmation avec Python

- (Les langages de programmation)
- La plateforme Python et l'IDE
- Les fondamentaux du langage Python
- Les fonctions et paramètres
- Les outils de débogage
- Les variables complexes
- L'essentiel de la POO (Programmation Orientée Objet)

Le développement web côté serveur avec ASP.Net Core

(Durée de 2 semaines)

- Présentation de la plateforme .NET
- Les bases du langage C#
- Le langage de requête LINQ
- ASP.NET MVC
- ASP.NET Razor
- Les contextes d'exécution et les cookies
- Entity Framework
- Authentification et autorisation
- Développer et consommer une API Rest
- Déployer une application ASP.Net Core