

Concepteur·rice Développeur d'Applications

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

Reconnu par l'État de niveau 6 (ou niveau BAC+4)
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 : 1 015 heures
- Stage pratique en entreprise : 280 heures
- Durée totale de la formation : 1 295 heures

En alternance :

- Formation en centre : 1015 heures
- Sur un contrat de 18 à 24 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 Imagine Premium et VMware Academic Program (licences logiciels Microsoft et VMware)
- 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, Struts, Hibernate, Tomcat, Java...)
- Systèmes de gestion de bases de données relationnelles (Oracle Database, 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-riche Développeur 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 détaillé

La Programmation Orientée Objet (POO) avec Java

- Les concepts de programmation orientée objet
 - Les classes
 - Les associations
 - L'héritage
 - Les classes et méthodes abstraites
 - Les interfaces
- La gestion des exceptions
- Les génériques
- Les collections
- Les énumérations
- Le diagramme de classes (UML)

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

Les notions complémentaires sur JAVA SE

- Rappeler les concepts essentiels de la POO
- Approfondir la notion d'Interface
- Approfondir la notion de Généricité
- Découvrir les expressions Lambda et l'API Stream
- Savoir organiser une application en s'appuyant sur une architecture en couches
- Intégrer la gestion des exceptions dans une architecture en couche
- Savoir tracer le fonctionnement de l'application au travers des log

Le développement Web Côté Serveur – Back-End avec Java Spring Boot

- Spring - Développer une application web classique organisée en couches :
 - Comprendre l'architecture d'une application web classique
- Créer et configurer l'application :
 - Comprendre la notion de moteur de production
 - Avantage du Framework Spring Boot
- Couplage fort/faible :
 - Design pattern Injection de dépendance et inversion de contrôle (IOC)
- Structurer l'application en couche avec Spring Core • Spring Web - Développer la couche IHM d'une application web :
 - Design pattern MVC
 - Le moteur de template Thymeleaf
 - Les contextes d'exécution o Les formulaires
 - La validation des données
 - L'internationalisation
- Spring JDBC - Développer la Couche d'accès aux données :
 - Situer la couche DAL, ses responsabilités
 - Connaitre l'API JDBC
 - L'utiliser avec Spring JDBC
 - Utilisation des Procédures stockées (ex. : TRANSACT SQL, ...)
 - Comprendre et mettre en place le pattern DAO
- Situer la couche BLL :
 - Couche BLL met en place le cœur du métier
 - Déterminer ses responsabilités
 - Respecter les cas d'utilisation
 - Bloquer les données invalides de la couche présentation
 - Bloquer les exceptions de la base de données
 - Comprendre la gestion des exceptions
 - Gérer les transactions
- Sécuriser une application web (authentification et autorisations) :
 - Spring Security
 - Mise en place de la sécurité
 - Authentification et Habilitation
 - Restriction sur l'accès des URLs
 - Limiter l'affichage selon les rôles
 - Gestion des exceptions

Projet N°2 : Web (Java Spring Boot)

- 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

Analyse et Conception (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, ...)

Javascript avancé - Initiation Framework Angular

- Apprendre les bases du javascript
- Savoir manipuler le DOM
- Tester du code JavaScript
- Connaitre le fonctionnement des objets en Javascript
- Appréhender le principe de prototype
- Utiliser la syntaxe ES6
- *Savoir ce qu'est une API et comment cela fonctionne*
- *Connaitre plus particulièrement le fonctionnement des API RestFul*
- *Savoir comment envoyer des requêtes asynchrones*
- Appréhender les bases du TypeScript
- Utiliser la POO avec Typescript
- Être capable d'utiliser les modules et génériques
- *Comprendre les bases d'Angular*
- *Utiliser des services avec Angular*
- *Contacteur une API*

Développement Javascript Server /Node.js et NoSQL

- Rappels sur le JavaScript
- Installation des outils
- Découverte de Node.js
- Utilisation du framework Express.js
- Base de données NoSQL avec MongoDB
- Les webservices avec Node.js et la documentation avec Swagger

Le développement Web côté Serveur – Back-End (PHP)

- Introduction au langage PHP
- Rappels et POO avec PHP
- Les formulaires en PHP :
 - Le traitement
 - La validation des données
- L'accès aux données :
 - Les requêtes non préparées
 - Les requêtes préparées
- Les échanges de données entre le client et le serveur :
 - Les redirections
 - Les sessions
 - L'appel au serveur (ex. : Ajax, ...)

Le développement Web côté Serveur – Back-End (Symfony)

- Introduction à Symfony
- Vues et Twig
- Routage et contrôleurs
- Formulaires • Accès aux données avec un ORM (Doctrine)
- Création de services (service métier)
- Utilisateurs et sécurité
- Fixtures
- Migration
- Déploiement d'une application
- Réalisation de tests avec PHPUnit
- Création de commandes personnalisées
- Les services Web

Projet N°3 : Web (Symfony)

- Mise en place de la base de données
- Mise en place de données de test
- Application d'une charte graphique définie préalablement
- Mise en œuvre d'une organisation de projet, estimation des tâches
- Utilisation d'outils de gestion du code et travail en équipe

Le développement d'une application mobile native (Kotlin)

- • Découverte de l'Environnement
- • Développement des Interfaces Graphiques
- • Kotlin (Init.Prog. + POO)
- • Envoyer des Données à l'IHM
- • Navigation Interne et Externe
- • Cycle de Vie
- • ViewModel avec LiveData
- • Room (ORM)
- • RecyclerView
- • Android Studio et Git
- • Communiquer avec une API REST
- • Firebase

Utilisation avancée du Framework JS : 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

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

Le SQL avancé (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

Les Frameworks de Java : Spring Security et ORM

- **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

DevOps - Infrastructure et déploiement d'applications

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

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)

L'Intelligence Artificielle avec Python

- Présentation du Machine Learning
- L'environnement technique
- Les bibliothèques Python pour le ML (NumPy, Pandas, ...)
- Présentation des Datasets
- Le travail sur les données
- L'apprentissage supervisé

L'IA avec Python + Projet final

- Les algorithmes de classification
- L'apprentissage non-supervisé
- Concepts avancés sur l'apprentissage
- **Projet (TP) final** : Petite compétition en mode Kaggle.
- **Bilan** : Les risques et les limitations de l'IA, les enjeux éthiques

Module optionnel (selon profil)

L'algorithmique

- Introduction à l'algorithmique
- Les instructions de base en pseudo-code
- Les instructions conditionnelles
- Les instructions itératives
- Les tableaux
- Les procédures et fonctions

Initiation à la programmation procédurale avec Java

- Prise en main de l'environnement
- Les langages de programmation
- La plateforme Java
- Environnement de développement
- Les fondamentaux du langage
- Les méthodes et les paramètres
- Les outils de débogage
- Les variables complexes

Le développement Web côté Client – Front-End (HTML et CSS)

- Le Web et le client léger
- Les balises HTML
- Les balises multimédia
- Navigation
- Les feuilles de style CSS
- Unités de mesure
- Position des éléments graphiques (CSS)
- Le Responsive Web Design (Media Queries)
- Les formulaires HTML
- Un Framework CSS (ex. : Bootstrap, ...)
- Les préprocesseurs (ex. : SASS, ...)
- Maquettage et design (ex. : Figma, Adobe XD, ...)

Initiation à Javascript

- Introduction au langage
- Contrôler le DOM avec Javascript
- Manipulation de formulaires
- Développer des bibliothèques
- Utiliser une bibliothèque du marché (ex. JQuery, ...)

Projet N°1 : Web (HTML, CSS, JS)

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

Option facultative

L'application ASP.Net Core (via CEL)

- 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

Projet final : Web (.Net Core)

- Projet Final : au choix avec objectif de le présenter