

# CEFISI – 2019/2020

## Parcours « Concepteur développeur Java/JEE » Titre RNCP niveau II

### Publics concernés

Public en reconversion à haut potentiel, inscrit à Pôle Emploi,

### Prérequis

Expérience significative en développement.

Rigueur du raisonnement logique, faculté d'analyse, esprit de synthèse, sens de l'organisation, facilités de communication et d'expression, capacités d'autonomie et d'initiative.

Connaissance en anglais notamment technique (lire, comprendre des textes courts, poser des questions simples sans trop d'hésitation et exprimer des opinions).

### Objectif

Ce parcours de formation, adossé à la [certification RNCP de niveau 2 « Concepteur développeur informatique »](#), a pour objectif que les apprenants puissent développer de façon professionnelle des applications d'entreprise (Web et mobiles) en Java, en couvrant tout le cycle de développement et de maintenance, de la spécification à la recette.

Les compétences visées comprennent en particulier :

1. [Modéliser](#) l'application, notamment avec [UML](#), et spécifier en même temps les [tests fonctionnels](#) permettant de valider l'application.
2. [Concevoir](#) l'application et son [interface utilisateur](#), en faisant les [choix techniques](#) les plus appropriés.
3. [Programmer](#) l'application, en même temps que ses [tests unitaires](#) associés.
4. Créer des [jeux d'essais](#) pour tester l'application et son [intégration](#) au système existant.
5. [Déployer](#) l'application pour une mise en production.
6. [Documenter](#) l'application, notamment via la [javadoc](#).

7. Employer les **méthodes agiles**, tant côté technique que côté organisation.

La certification RNCP se compose précisément de 3 compétences :

1. **Développer la persistance des données**
  1. Concevoir une base de données
  2. Mettre en place une base de données
  3. Développer des composants dans le langage d'une base de données
  4. Utiliser l'anglais dans son activité professionnelle en informatique.
2. **Développer des composants d'interface**
  1. Maquetter une application
  2. Développer une interface utilisateur
  3. Développer des composants d'accès aux données
  4. Développer des pages Web en lien avec une base de données
3. **Développer une application n-tiers**
  1. Concevoir une application
  2. Collaborer à la gestion d'un projet informatique
  3. Développer des composants métier
  4. Construire une application organisée en couches
  5. Développer une application de mobilité numérique
  6. Préparer et exécuter les plans de tests d'une application
  7. Préparer et exécuter le déploiement d'une application

## Méthodes et moyens pédagogiques

Les cours, consacrés à l'acquisition des savoirs, savoir-faire et savoir-être, sont encadrés par un **formateur spécialiste du sujet**. Les projets, consacrés à la consolidation des acquis, alternent phases en autonomie et phases supervisées, où le formateur donne un retour, validant ce qui va dans le sens attendu, réorientant ce qui doit être rectifié, et aplanissant les difficultés rencontrées. En autonomie, les apprenants peuvent également faire appel aux formateurs présents, selon leur disponibilité, et à l'administration.

La pédagogie accorde une très grande place à la **pratique**, indispensable pour une réelle acquisition des savoir-faire et savoir-être. Selon le cours, celle-ci prend la forme d'exercices techniques individuels (ex : bases de données), d'études de cas (ex : UML), de travail en groupes (ex : projet). Les formateurs sont invités à faire appel aux **jeux coopératifs**, aux **outils coopératifs** (ex : rédaction collective, wiki) et aux applications ad hoc (**cartes heuristiques**, **outils** de gestion et de suivi en ligne ou bureautiques).

Les **méthodes agiles**, introduites progressivement dès le premier jour, sont employées tout au long de la formation, tant côté technique (tests unitaires, gestion de version, intégration continue, outils des environnements de développement) que côté organisation (cycles courts, travail par binôme, travail par itération, management visuel, « poker planning », etc.).

Les exposés visent à faire **comprendre en profondeur** les sujets traités, et font appel aux **retours d'expérience** des formateurs qui veillent à transmettre l'état de l'art et les bonnes pratiques.

## Méthodes et outils d'évaluations

## Avant l'entrée en formation

Nous évaluons systématiquement lors d'un **entretien** l'adéquation de nos formations avec le profil et le projet professionnel des candidats : nous tenons à ce qu'elles les mènent à un emploi pérenne dans lequel ils s'épanouissent, et qu'ils soient pleinement engagés, la motivation étant un facteur clé de leur succès et de la cohésion du groupe formé.

Pour ce parcours, nous veillons en particulier à la rigueur, la curiosité, le sens du détail, le goût de l'ingénierie, et l'aptitude au travail en équipe

Le parcours, riche et dense, nécessite un prérequis que nous évaluons via *3 tests d'une durée totale de 1 heure* :

- Test de **logique** et d'**abstraction** : le développement logiciel nécessite de très bonnes capacités logiques et d'abstraction.
- Test de **programmation** : savoir déjà écrire des programmes doit déjà être acquis.
- Test d'**anglais** écrit : l'anglais est incontournable en développement, la documentation et les meilleurs forums techniques sont en anglais, qui de plus est souvent la langue de travail à l'écrit.

## Durant la formation

Les évaluations en cours de formation ont pour objet de fournir un retour aux apprenants, afin de les encourager et leur indiquer comment progresser. Elles sont également un *élément indicatif de l'évaluation finale*.

Les évaluations de compétences et savoir-être se fait via la pratique (exercices et surtout projets), et prend la forme d'un retour indiquant aux apprenants leurs **acquis**, et les alertant sur les **points à améliorer**. Elles sont ainsi proches des évaluations en entreprise.

Les connaissances sont évaluées via des **questionnaires**, notamment à choix multiples.

## Après la formation

La formation est validée par la certification RNCP de niveau 2 (bac+3/4) « Concepteur développeur informatique ».

La certification consiste en une soutenance de 1h20 face à un jury professionnel indépendant du centre de formation. Les candidats remettent préalablement au jury un dossier de synthèse décrivant à partir d'exemples concrets leurs pratiques professionnelles visées dans le titre, ainsi qu'un dossier contenant les livrables clés d'un projet couvrant toutes ces activités. Les candidats présentent oralement ce projet, avec un support de type powerpoint, et sont ensuite interrogés par le jury.

Le jury s'appuie sur le dossier fourni par le (la) candidat(e), la soutenance et les évaluations réalisées pendant le parcours de formation.

Le parcours de formation comprenant un projet de synthèse en équipe de 13 jours, les candidats peuvent se présenter à la certification peu après la fin de la formation. Il est toutefois préférable



de la passer après une période de 3 mois en entreprise, qui permet de renforcer les acquis et de les inscrire dans la durée.

## Durée (en jours et heures)

La formation dure 57 jours (400 h).

## Supports pédagogiques

Les apprenants ont à disposition sur l'intranet du centre les **supports** de chaque cours, des liens vers une **bibliographie** (livres ou sites Web), ainsi que les énoncés des **exercices**. Lorsqu'il est pertinent de fournir une **solution** (ex : modèle de données répondant à un besoin fonctionnel), celle-ci est disponible une fois la correction effectuée. Le code source et la documentation de projets exemples ou de projets passés est disponible sur l'intranet ou sur les plateformes collaboratives comme *github*.

Nous disposons également d'une **bibliothèque** de plus de 50 ouvrages consacrés aux disciplines enseignées, qui peuvent être empruntés par les apprenants, après signature d'un chèque de caution non encaissé et remis à l'apprenant à la fin de la formation.

Enfin, dans notre domaine où l'environnement évolue constamment et très vite, l'une des sources les plus utilisées est le Web. Nous indiquons à nos apprenants des liens utiles, qu'ils peuvent explorer et compléter par leurs propres recherches grâce à notre connexion **Internet via wifi à haut débit**.

## Conditions d'organisation matérielle

Le CEFISI dispose de 2 grandes salles pouvant accueillir confortablement 16 personnes (voir photos ci-jointes), dotées chacune d'un tableau blanc, d'un vidéo-projecteur et d'un paper board.

Un réseau wifi avec répéteur assure une connexion Internet permanente à haut débit. Les apprenants disposent d'un intranet où sont stockés les supports de cours et les exercices.

Les apprenants utilisent leur propre ordinateur portable, et disposent ainsi du même environnement de travail au centre et chez eux. Pour ceux qui n'ont pas de portable, le centre leur en prête un durant les cours.

Les locaux sont accessibles aux personnes à mobilité réduite, et satisfont aux critères de sécurité incendie. Des extincteurs sont notamment disponibles. Par ailleurs, Mme Oubekkou est formée au secourisme.

Un espace repas, doté de micro-ondes et d'un frigo, est également disponible. Les apprenants peuvent apporter leur repas, le mettre au frigo et le réchauffer avec les micro-ondes. Un distributeur de boissons est disponible dans l'entrée.

## Lieux d'intervention possible sur le territoire concerné

Les formations ont lieu à l'adresse du CEFISI : 1 voie Félix Eboué - 94000 Créteil, à 4mn à pied du métro Créteil l'Echat, et à 1mn du terminus de la ligne 172 (Hôpital Henri Mondor).

Téléphone

01.48.98.94.82

Mail

contact@cefisi.com

Web

www.cefisi.com

Siret

81743947400013

Code APE

8559A

## Positionnement

Ce parcours de formation est centré sur les **technologies Java**, langage le plus utilisé pour les applications (**Web, mobiles et distribuées**) dans les grandes entreprises.

Il est conçu pour répondre aux besoins de ces entreprises et tout particulièrement des **ESN**, qui recherchent des professionnels compétents dans ces **technologies complexes, attentifs au besoin**, et sachant **travailler en équipe**, par **projet**. En particulier, elles attendent d'eux une maîtrise des méthodes de développement, en particulier des **méthodes agiles**, et des cadres (**frameworks**) incontournables sur le marché.

La somme de connaissances et de compétences à acquérir est très importante. Pour cela, nous installons des **fondamentaux** solides, par **paliers**, en intégrant constamment méthode, outils et technologies, à travers des **projets**, individuels ou en équipe, tels le développement d'une application Web de gestion de projets, une plateforme Web et mobile de conciergerie d'entreprises, etc.

## Programme détaillé

Pour éviter de les répéter dans chaque module, nous indiquons ici les critères d'évaluation, communs à la plupart de ces modules :

- Qualité externe (perçue par l'utilisateur) : couverture fonctionnelle du projet ou des exercices, absence (ou faible taux) de bugs, qualité des messages d'erreur (précis et informatifs), clarté et simplicité de l'interface utilisateur
- Qualité interne (perçue par les développeurs) : organisation des classes, concision et simplicité des fonctions et procédures, respect des normes de nommage, clarté et pertinence des architectures, exactitude des modèles de données, généricité des solutions, utilisation pertinente des schémas de conception, absence d'éléments superflus.
- Qualité de la méthode : simplicité, utilité et couverture des tests unitaires, utilisation de l'outil de gestion de version, développement par cycles courts, concevoir avant de programmer, constitution et utilisation de jeux d'essai, revue de code.

## Conception et programmation objet avec Java (5 jours)

<b>Objectif</b>	Comprendre les concepts de la programmation objet et savoir comment concevoir par objets
<b>Prérequis</b>	Pratique d'un langage de programmation
<b>Contenu</b>	<ul style="list-style-type: none"> <li>• Répartir les traitements d'une application :             <ul style="list-style-type: none"> <li>○ Classes comme types et comme modules</li> <li>○ Encapsulation : vue externe d'une classe (services), vue interne (méthodes et attributs), niveaux d'accès dans les différents langages</li> <li>○ Conception par contrat : types abstraits, préconditions, postconditions et invariants de classe.</li> <li>○ Exceptions</li> <li>○ Séparation modèle/interface utilisateur</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>• Regrouper le code :             <ul style="list-style-type: none"> <li>○ Héritage, polymorphisme, typage statique, liaison dynamique</li> <li>○ Classes abstraites et interfaces</li> <li>○ Généricité. Collections.</li> </ul> </li> <li>• Utiliser des outils agiles             <ul style="list-style-type: none"> <li>○ Tests unitaires avec JUnit</li> <li>○ Gestion de version</li> </ul> </li> </ul>
<b>Méthode</b>	Tous les concepts étudiés sont mis en pratique immédiatement à travers des exercices progressifs. Nous insistons beaucoup sur la méthode : comment commencer, quelles questions se poser, comment tester. Nous comparons les langages Java, C# et Eiffel.
<b>Evaluation</b>	Test

## SQL (2 jours)

<b>Objectif</b>	Comprendre le modèle relationnel, son apport, et savoir interroger les bases de données relationnelles avec SQL.
<b>Prérequis</b>	Aucun
<b>Contenu</b>	<ul style="list-style-type: none"> <li>• Fonctions d'un système de gestion de bases de données (SGBD). Comparaison de Oracle, SQL Server, MySQL, Access</li> <li>• Le modèle relationnel : tables, relations, champs, enregistrements, clés primaires, contraintes d'intégrité, clés étrangères, opérateurs relationnels, transactions.</li> <li>• Modification des données avec INSERT, UPDATE, DELETE</li> <li>• Lecture des données avec SELECT. Utilisation de DISTINCT.</li> <li>• Requêtes imbriquées retournant une valeur, n valeurs et n lignes.</li> <li>• Agrégations, groupements (GROUP BY), restrictions sur groupement (HAVING).</li> <li>• Utilisation de MysqlWorkbench et phpMyAdmin</li> </ul>
<b>Méthode</b>	Exposé et mise en pratique immédiate avec des exercices sur machine avec une base de données fournie.
<b>Evaluation</b>	Couverture des exercices proposés

## Java SE (3 jours)

<b>Objectif</b>	Savoir développer des clients lourds en Java
<b>Prérequis</b>	<ul style="list-style-type: none"> <li>• Java (module conception et programmation objet)</li> <li>• SQL</li> </ul>
<b>Contenu</b>	<ul style="list-style-type: none"> <li>• Séparation modèle/interface utilisateur.</li> <li>• Composants d'interface.</li> <li>• Principe de la programmation événementielle.</li> </ul>

	<ul style="list-style-type: none"> <li>• La bibliothèque jdbc : établir une connexion, exécuter des requêtes SQL, parcourir des résultats.</li> <li>• Gestion des exceptions, récupération des erreurs SQL.</li> <li>• Test unitaire du modèle.</li> <li>• Entrées-sorties</li> </ul>
<b>Méthode</b>	Exposé immédiatement mis en pratique via un mini projet. Ressources en ligne.
<b>Evaluation</b>	Au vu de la réalisation

## Android (4 jours)

<b>Objectif</b>	Savoir développer des applications Android simples
<b>Prérequis</b>	<ul style="list-style-type: none"> <li>• Java (module conception et programmation objet)</li> <li>• SQL</li> <li>• Java SE</li> </ul>
<b>Contenu</b>	<ul style="list-style-type: none"> <li>• Mise en place de l'environnement</li> <li>• Gestion des activités</li> <li>• Les ressources et l'interface graphique</li> <li>• Gérer le cycle de vie des applications</li> <li>• Persistance avec SQLite</li> <li>• Gestion des permissions</li> </ul>
<b>Méthode</b>	Exposé immédiatement mis en pratique via un mini projet. Ressources en ligne.
<b>Evaluation</b>	Au vu de la réalisation

## Conception et programmation des bases de données (4 jours)

<b>Objectif</b>	Savoir concevoir des bases de données assurant l'intégrité de leurs données, à partir d'une expression de besoin fonctionnel. Savoir les programmer avec des procédures stockées et des déclencheurs pour normaliser les données et garantir leur intégrité.
<b>Prérequis</b>	SQL
<b>Contenu</b>	<ul style="list-style-type: none"> <li>• Formes normales</li> <li>• Modèle entités-associations. Notations UML et Merise.</li> <li>• Utilisation de mysqlworkbench : génération de la base depuis le schéma, génération du schéma depuis la base, synchronisation des deux.</li> <li>• Contraintes d'intégrité déclaratives.</li> <li>• Concurrence et gestion des transactions.</li> <li>• PL/SQL.</li> <li>• Déclencheurs.</li> </ul>

	<ul style="list-style-type: none"> <li>• Procédures stockées.</li> </ul>
<b>Méthode</b>	Exposé immédiatement mis en pratique via des exercices progressifs. Ressources en ligne.
<b>Evaluation</b>	Au vu de la réalisation

## UML (3 jours)

<b>Objectif</b>	Savoir comment modéliser avec UML
<b>Prérequis</b>	Module conception et programmation objet
<b>Contenu</b>	<ul style="list-style-type: none"> <li>• Vue d'ensemble d'UML, cohérence entre les diagrammes.</li> <li>• Modélisation fonctionnelle. Cas d'utilisation, nominaux, alternatifs et en cas d'échec. Relations entre cas (inclusion, héritage).</li> <li>• Modélisation dynamique. Diagramme de séquence, diagramme d'activité, diagramme d'états.</li> <li>• Modélisation statique. Diagramme de classes.</li> <li>• La conception par contrat en UML, OCL (Object Constraint Language)</li> </ul>
<b>Méthode</b>	Exposé immédiatement mis en pratique via un mini projet. Ressources en ligne.
<b>Evaluation</b>	Etude de cas

## Conception d'IHM (2 jours)

<b>Objectif</b>	Avoir les bases pour produire des interfaces utilisateur ergonomiques
<b>Prérequis</b>	Aucun
<b>Contenu</b>	<ul style="list-style-type: none"> <li>• Ergonomie : structurer pour pallier la limite des <math>7 \pm 2</math> éléments, communication implicite, communication analogique, bruit et silence.</li> <li>• Maquettage d'IHM</li> <li>• Les spécificités du Web et du mobile</li> <li>• Feuilles de style CSS : principes (sélecteurs, directives, règles, supports) et principales directives</li> <li>• Web adaptatif (responsive Web design) : media queries</li> </ul>
<b>Méthode</b>	Exposé immédiatement mis en pratique, avec des exemples donnés. Ressources en ligne.
<b>Evaluation</b>	Au vu de la réalisation

## Java EE (4 jours)

<b>Objectif</b>	Savoir développer des applications Web classiques en Java
-----------------	---



<b>Prérequis</b>	<ul style="list-style-type: none"> <li>• Java SE</li> <li>• SQL</li> </ul>
<b>Contenu</b>	<ul style="list-style-type: none"> <li>• Servlets</li> <li>• JSP, JSTL et bibliothèques de balises</li> <li>• Découpage MVC1 et MVC2</li> <li>• Portées page, requête, session, application</li> <li>• Filtres</li> </ul>
<b>Méthode</b>	Réalisation d'une application Java EE typique, étape par étape. Support de cours, documentation en ligne.
<b>Evaluation</b>	Au vu de la réalisation

### Spring MVC (3 jours)

<b>Objectif</b>	Savoir utiliser Spring MVC pour développer des applications Web en MVC
<b>Prérequis</b>	<ul style="list-style-type: none"> <li>• Java EE</li> </ul>
<b>Contenu</b>	<ul style="list-style-type: none"> <li>• Inversion de contrôle</li> <li>• Contrôleurs Spring, correspondance avec URLs et méthodes, lecture de paramètres http</li> <li>• Formulaires : cycle de vie, bibliothèques de balises</li> <li>• Factorisation de code avec les ModelAttribute</li> <li>• Gestion des exceptions</li> </ul>
<b>Méthode</b>	Exposé immédiatement mis en pratique via un mini projet. Ressources en ligne.
<b>Evaluation</b>	Au vu de la réalisation

### Intégration continue (2 jours)

<b>Objectif</b>	Savoir intégrer les développements de façon continue
<b>Prérequis</b>	<ul style="list-style-type: none"> <li>• Java EE</li> </ul>
<b>Contenu</b>	<ul style="list-style-type: none"> <li>• Configuration des outils</li> <li>• Rapports de tests unitaires</li> <li>• Envoi de mails de notification</li> <li>• Déploiement automatique</li> <li>• Rapports d'analyse qualité</li> </ul>
<b>Méthode</b>	Exposé immédiatement mis en pratique. Ressources en ligne.
<b>Evaluation</b>	Au vu de la réalisation

## Hibernate/JPA (4 jours)

<b>Objectif</b>	Savoir utiliser le framework Hibernate/JPA pour développer la couche de persistance
<b>Prérequis</b>	<ul style="list-style-type: none"> <li>• Java EE</li> </ul>
<b>Contenu</b>	<ul style="list-style-type: none"> <li>• La correspondance objet/relationnel : représentation en objet des relations 1-1, 1-n, n-n et relations d'héritage</li> <li>• Cycle de vie des objets, synchronisation avec la BD, persistance en cascade</li> <li>• Langage HQL</li> <li>• Transactions avec Hibernate</li> <li>• Gestion du cache</li> </ul>
<b>Méthode</b>	Exposé immédiatement mis en pratique via un mini projet. Ressources en ligne.
<b>Evaluation</b>	Au vu de la réalisation

## Services Web et Ajax (3 jours)

<b>Objectif</b>	Savoir concevoir et programmer des services Web REST
<b>Public</b>	Développeurs, chefs de projet
<b>Prérequis</b>	<ul style="list-style-type: none"> <li>• Java EE</li> </ul>
<b>Contenu</b>	<ul style="list-style-type: none"> <li>• Principes d'architecture des services Web. REST : ressources, interface uniforme (méthodes DELETE, GET, HEAD, OPTIONS, POST, PUT), codes retour http.</li> <li>• Conception de ressources Web.</li> <li>• Tests de ressources via JUnit.</li> <li>• Programmation de services Web avec spring ou jersey.</li> <li>• Appels de services Web via Ajax.</li> </ul>
<b>Méthode</b>	Exposé immédiatement mis en pratique via un mini projet. Ressources en ligne.
<b>Evaluation</b>	Au vu de la réalisation

## Anglais (3 jours répartis en 6 séances)

<b>Objectif</b>	Renforcer sa compétence en anglais technique oral et écrit
<b>Contenu</b>	<ul style="list-style-type: none"> <li>• Présentation orale d'un sujet technique</li> <li>• Rédaction de email et de compte rendu</li> <li>• Synthèse de documents</li> </ul>

<b>Méthode</b>	Alternance oral/écrit
<b>Evaluation</b>	Au vu du rendu et de la prestation orale

## Coaching appliqué à la recherche d'emploi (2 jours)

Ce module a pour objet de mobiliser les ressources des apprenants dans leur recherche d'emploi pour qu'elle soit le plus efficace possible. Le travail est personnalisé, et se prolonge durant tout le parcours, les apprenants ayant accès à un coach spécialisé.

<b>Objectif</b>	Développer sa capacité à communiquer efficacement dans sa recherche d'emploi : faire passer ses messages, comprendre ceux des autres et leur comportement
<b>Public</b>	Tout public
<b>Prérequis</b>	Aucun
<b>Contenu</b>	<ul style="list-style-type: none"> <li>• Déterminer ses points forts et ses motivations.</li> <li>• Les mettre en valeur dans son cv et ses lettres de motivation.</li> <li>• Recadrer ses points faibles pour être à l'aise avec.</li> <li>• Savoir se mettre dans un état de ressource lors d'un entretien.</li> </ul>
<b>Méthode</b>	Travail en groupe, questions-réponses, jeux de rôles, travail sur la voix, la respiration, la posture. Suivi individuel tout le long du cursus.
<b>Evaluation</b>	N/A

## Projet de synthèse (13 jours)

<b>Objectif</b>	Consolider les compétences acquises via un projet mené suivant les méthodes agiles
<b>Prérequis</b>	<ul style="list-style-type: none"> <li>• Java EE</li> <li>• UML</li> <li>• Intégration continue</li> <li>• Services Web et Ajax</li> <li>• Conception d'IHM</li> </ul>
<b>Contenu</b>	Analyse du besoin, conception, programmation, test et recette d'une application développée suivant les méthodes agiles, avec une soutenance intermédiaire.
<b>Méthode</b>	Projet en partie tutoré, suivant les méthodes agiles et notamment SCRUM
<b>Evaluation</b>	Soutenance finale