

Pourquoi cette UE ?

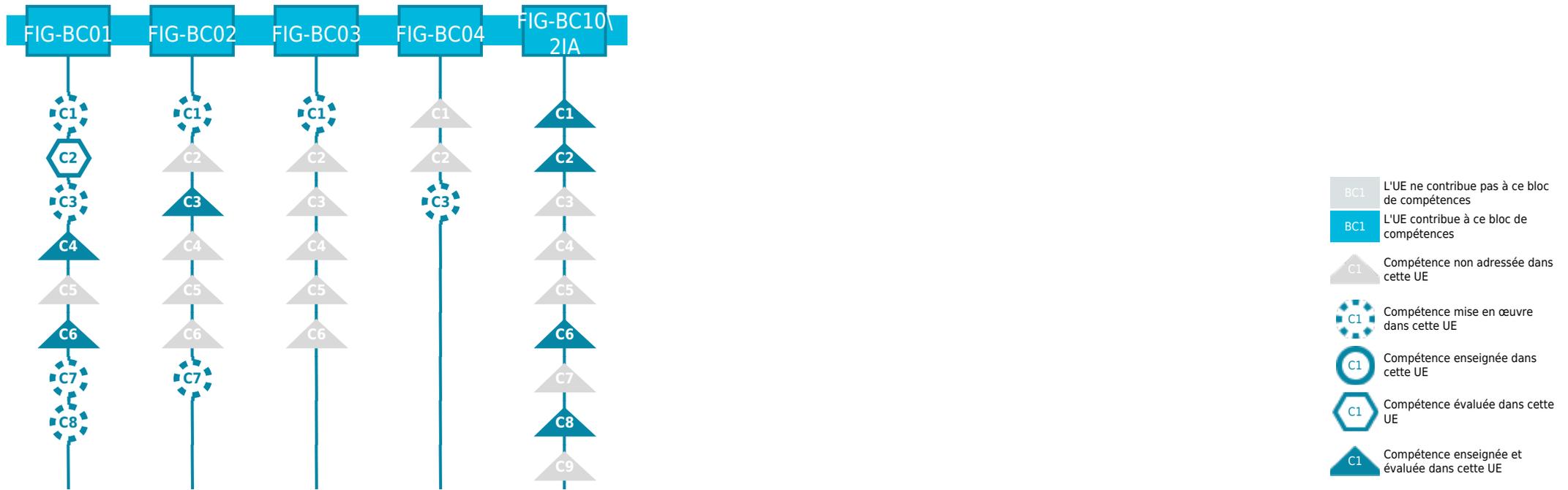
Les techniques de développement d'applications ont évolué pour répondre à des exigences de qualité et de productivité toujours plus importantes. La réutilisation est une solution concrète à cette problématique. Ce cours présente un ensemble de techniques permettant de réaliser ces interactions client-serveur entre applications distribuées, depuis les échanges de données (sockets) jusqu'aux composants et services métier (serveurs applicatifs et architectures n-tiers) en passant par les appels distants de fonctionnalités (middlewares). Il aborde également des notions de programmation Web avancées.

Éléments constitutifs de l'UE

	coefficient	
2IAiail_9_6-1 Client-serveur	1	
2IAiail_9_6-2 Architectures n-tiers	1	
2IAiail_9_6-3 Développement Web avancé	1	
Volume d'heures d'enseignement encadré	Volume d'heures de travail personnel	Nombre d'ECTS
62	44	4

Alignement curriculaire

Parmi les compétences visées par la formation, lesquelles sont développées dans cette UE ?



Contexte et enjeux de l'enseignement

Ce cours expose les concepts de la programmation distribuée et le développement d'architectures client-serveur. Il présente les différentes techniques de mise en œuvre actuelles, des plus simples, qui servent de couches basses, aux plus élaborées, qui permettent la mise en œuvre d'architectures et de mécanismes de gestion sophistiqués : - les sockets, permettant les échanges de données, - les middleware, permettant l'invocation distante de fonctionnalités et les échanges d'objets Il aborde les variations des architectures client-serveur.

Prise en compte des dimensions socio-environnementales

ODD9 - Industrie, innovation et infrastructure

Prérequis

Programmation en langage Java

Modalités d'enseignement et d'évaluation

	Nb d'heures
Cours	6
Cours intégré (cours + TD)	
TD	0
TP	0
Projets	14
Travail en autonomie encadré	
Contrôles et soutenances	0
Travail personnel	15

Objectifs pédagogiques

(à la fin de cet enseignement, l'étudiant sera capable de ...)

- compréhension des architectures client-serveur
- développement d'applications distribuées à l'aide de sockets
- développement de composants distribués à l'aide d'un middleware

Activités

(CM, TD, TP, projet, sortie terrain, etc.)

Les enseignements sont prévus pour 30 élèves. Les TP seront réalisés sur les ordinateurs personnels de ces derniers.
Le découpage est prévu comme suit :
- 6h de cours
- 9h de projet (+ travail personnel)

Évaluations et retours faits aux élèves

(évaluations qui comptent pour la note ou qui permettent à l'étudiant de se situer, corrigés, feedback personnalisé...)

Evaluation sur projet (coef 1)

Retour sur l'évaluation fait à l'élève : retours aux élèves pendant la soutenance.

2IAiail_9_6 Programmation distribuée

FIG

2IAiail_9_6-1 Client-serveur

S9

Plan de cours

- Principes des architectures C/S
- Communication sous IP (sockets)
- Serveurs multi-tâches
- Remote Procedure Call
- Formes d'architecture C/S
- Réutilisation et composants logiciels
- Bus logiciel et composants distribués
- Mise en place d'une application serveur RMI
- Mise en place d'une application cliente RMI

Ressources et références

1 Polycopié

Contexte et enjeux de l'enseignement

Ce cours expose les concepts de la programmation d'architectures logicielles multi-tiers, c'est-à-dire composées d'applications clientes utilisant les services fournis par des serveurs applicatifs interagissant avec des bases de données. Il présente, au travers de JEE, un exemple de framework facilitant le développement de ce type d'architectures. Il aborde également le développement de clients lourds (applications distantes) ou légers (pages web dynamiques).

Prise en compte des dimensions socio-environnementales

ODD9 - Industrie, innovation et infrastructure

Prérequis

- programmation en langage HTML - programmation en langage JAVA - programmation de Bases de Données (langage SQL)

Modalités d'enseignement et d'évaluation

	Nb d'heures
Cours	6
Cours intégré (cours + TD)	
TD	0
TP	0
Projets	9
Travail en autonomie encadré	0
Contrôles et soutenances	0
Travail personnel	9

Objectifs pédagogiques

(à la fin de cet enseignement, l'étudiant sera capable de ...)

- développement d'architectures n-tiers avec le framework JEE
- développement de clients légers (web) avec les technologies Java
- développement de clients lourds (middleware) avec les technologies Java
- développement de mappings objet-relationnel avec JPA et Hibernate

Activités

(CM, TD, TP, projet, sortie terrain, etc.)

Les enseignements sont prévus pour 30 élèves. Les TP seront réalisés sur les ordinateurs personnels de ces derniers.
Le découpage est prévu comme suit :
- 6h de cours
- 12h de projet (+travail personnel)

Évaluations et retours faits aux élèves

(évaluations qui comptent pour la note ou qui permettent à l'étudiant de se situer, corrigés, feedback personnalisé...)

Projet (coef. 1)

Retour sur l'évaluation fait à l'élève : retours aux étudiants pendant la soutenance.

Plan de cours

- Infrastructures n-tiers (Java Enterprise Edition)
- Tiers présentation : technologies Java (JSP, Servlet, serveur web Tomcat pour le développement de clients légers et RMI/JNDI pour les clients lourds)
- Tiers métier : composants EJB et serveurs applicatifs Java Enterprise
- Tiers-système d'information : gestion du mapping objet-relationnel (JPA, Hibernate)

Ressources et références

1 Polycopié

Contexte et enjeux de l'enseignement

Ce cours a pour objectif d'expliquer les principes avancés de développement des applications Web, qui sont les applications les plus développées actuellement. La modularité d'une application web, maîtrisée grâce aux outils et technologies de développement actuels, permet de faire interagir des personnes aux compétences multiples dans le cadre d'un processus agile.

Prise en compte des dimensions socio-environnementales

ODD9 - Industrie, innovation et infrastructure

Prérequis

Module 2IA-il 8.5 (Développements Web et Mobile)

Modalités d'enseignement et d'évaluation

	Nb d'heures
Cours	5
Cours intégré (cours + TD)	
TD	
TP	20
Projets	0
Travail en autonomie encadré	0
Contrôles et soutenances	2
Travail personnel	20

Objectifs pédagogiques

(à la fin de cet enseignement, l'étudiant sera capable de ...)

Connaissance des différentes couches logicielles d'une architecture web fullstack (frontend et backend) et des techniques de développement associées. Chaque évolution à venir des techniques de développement va ajouter de nouvelles couches, de nouveaux frameworks ou bibliothèques. L'étudiant doit être capable de comprendre l'impact de ces nouveautés sur son application Web de référence.

Activités

(CM, TD, TP, projet, sortie terrain, etc.)

Les enseignements sont prévus pour 30 élèves. Les TP seront réalisés sur les ordinateurs personnels de ces derniers. Après une démonstration des technologies utilisées dont un framework back-end de type Spring.

Évaluations et retours faits aux élèves

(évaluations qui comptent pour la note ou qui permettent à l'étudiant de se situer, corrigés, feedback personnalisé...)

Projet personnel rendu en temps limité de 2h similaire à celui réalisé pendant les TP

Retour sur l'évaluation fait à l'élève : 3 semaines après la dernière séance.

Plan de cours

Le cours propose de s'initier aux concepts proposés par le framework JEE pour développer des applications WEB n-tiers respectant le patron de conception MVC (modèle-vue-contrôleur). La couche de présentation de cette architecture n-tiers est réalisée par une combinaison de technologies reconnues de la pile HTML/CSS/Javascript. La partie Javascript sera ensuite étendue en exploitant des frameworks récents tel que JQuery/Angular/ReactJs. De la même façon, la couche CSS sera exploitée au travers de Bootstrap CSS dans l'esprit du design « CSS Zen Garden ». Pour la couche backend, on exploitera les tests unitaires Junit et les logger log4j pour garantir la nonrégression de notre application. Une couche de persistance de type Dao-JDBC sera implantée pour démontrer la pertinence de la séparation avec les classes POJO définies. L'apprentissage se fait au travers d'une application Web créée par ajouts successifs à chaque Cours/TP. Les outils open source utilisés sont installés et paramétrés sur l'ordinateur personnel des étudiants au cours de la première séance de TP.

Ressources et références

Polycopié et site internet <https://pierrejean.wp.imt.fr/category/imt-mines-ales/>