

Année universitaire 2024/2025

MIAGE Informatique pour la Finance - 2e année de Master

Crédits ECTS : 60

LES OBJECTIFS DE LA FORMATION

Le parcours MIAGE-IF a pour objectif de former des informaticiens spécialistes des applications financières, capables de maîtriser tout au long de leur carrière les problématiques spécifiques à ce domaine et aptes à proposer et à mettre en œuvre les solutions adéquates. La formation est à 75% une formation dédiée aux connaissances nécessaires à l'ingénierie d'applications de haute qualité.

Les objectifs de la formation :

- Avoir des connaissances approfondies des technologies de pointes pour la conception de système d'information
- Former des informaticiens, dotés de solides compétences conceptuelles et technique
- Maîtriser la conception et de la réalisation d'applications exigeantes en termes de sécurité, de performance ou de complexité architecturale

PRÉ-REQUIS OBLIGATOIRES

- Titulaires d'un diplôme BAC+4 (240 crédits ECTS) ou équivalent à Dauphine, d'une université ou d'un autre établissement de l'enseignement supérieur en informatique
- Etudiants issus d'une formation professionnalisée (IUP, MST ou école d'ingénieurs), et ou disposant d'une expérience professionnelle significative en informatique

POURSUITE D'ÉTUDES

Formé en alternance, vous intégrez rapidement un secteur demandeur. Les entreprises partenaires du Masters sont :

- Grands comptes dans le domaine financier (banques et assurances notamment)
- Société de services et de conseil dans le domaine de l'informatique financière
- Editeurs de logiciels financiers

Débouchés : Analyste développeur, Auditeur financier, Business analyst, Chargé(e) de gestion des risques, Consultant(e), Ingénieur d'études, MOA Finance.

Ces différents métiers se situent dans la même sphère fonctionnelle et il est très courant de passer de l'un à l'autre au cours d'une carrière. La formation assurée dans le parcours MIAGE-IF en apprentissage a pour but d'assurer une adaptation facile à ces différents métiers et doit permettre à ses diplômés d'apporter des solutions innovantes par la combinaison d'une double compétence et d'une maîtrise des technologies les plus récentes.

PROGRAMME DE LA FORMATION

- Semestre 3
 - UE Obligatoires
 - Systèmes distribués et nouvelles technologies Web en Java
 - Actuariat pour le crédit et l'assurance

- Finance des marchés 1
- Finance entrepreneuriale
- Gestion de portefeuille et risque
- Management de Projets
- Culture générale en finance
- Les fondamentaux de la gestion d'actifs numériques
- Finance des marchés 2
- Systems, Languages and Paradigms for Big Data
- Semestre 4
 - UE Obligatoires
 - Anglais
 - Applications Web Orientées Services
 - C++
 - Machine Learning pour la finance
 - Méthodes Agiles d'Ingénierie Logicielle
 - Projet Programmation financière
 - Blockchains
 - SQL, NoSQL et NewSQL
 - Conduite de projet de communication
 - Gouvernance des systèmes d'informations bancaires
 - Livret d'entreprise et apprentissage
- Bloc Mémoire
 - UE Obligatoires
 - Mémoire

DESCRIPTION DE CHAQUE ENSEIGNEMENT

Actuariat pour le crédit et l'assurance

ECTS : 3

Description du contenu de l'enseignement :

Initier les étudiants à l'actuariat.

Actuariat des produits de taux simples en univers déterministe.

Obligations, zéro coupon.

Evaluation, courbe de taux, taux futurs, calcul du taux de rendement interne et de la valeur actuelle nette.

Quelques exemples de modèles stochastiques pour aller plus loin.

Principes de l'actuariat en assurance vie et IARD : tables de mortalité et de sinistre, calcul de prime quand les flux sont aléatoires.

Exemples d'implémentation en Python.

Compétence à acquérir :

Connaissance générale en actuariat. Evaluation d'obligations, courbe de taux, calcul de prime en assurance.

Anglais

ECTS : 3

Description du contenu de l'enseignement :

Cours par niveau

Applications Web Orientées Services

ECTS : 3

Description du contenu de l'enseignement :

Introduction aux microservices, constituant un paradigme d'architecture logicielle visant décomposer une système applicatif en une suite de services indépendants et découplés. Ce type d'architecture a émergé de la nécessité de construire des systèmes applicatifs à l'échelle du Web, c'est à dire continuellement évolutifs avec des niveaux de performance et de résilience jamais atteints.

- Les concepts et les enjeux des architectures microservices :
 - * Enterprise-scale IT à Web-scale IT (Cloud : IaaS, PaaS, SaaS, agilité, scalabilité horizontale, "design for failure"....)
 - * Architectures SOA traditionnelle versus Microservices
 - * Décomposition d'une application monolithique en microservices

- Les technologies et les pratiques de développements associées
 - * Architecture REST, API
 - * Microservices patterns
 - * Conteneurisation des applications (Docker, Kubernetes,...)
 - * Supervision et traces (Elastic Stack,...)

Mise en oeuvre dans le cadre d'un projet Java avec Spring Boot, Spring Cloud

Blockchains

ECTS : 1.5

C++

ECTS : 3

Description du contenu de l'enseignement :

Le langage C++ est actuellement le 3e langage de programmation le plus utilisé au monde (TIOBE Programming Community Index).

Ce cours est destiné en priorité aux étudiants débutants en programmation C++.

Classes et objets (constructeurs/destructeur, Propriétés des méthodes, Surcharge des opérateurs, Objet membre)

Héritage (simple, multiple, virtuel)

Patron de fonctions et Patron de classes Pré requis : Langage C et programmation orientée-objet

Compétence à acquérir :

Former les étudiants à la programmation orientée-objet en C++.

Conduite de projet de communication

ECTS : 1

Description du contenu de l'enseignement :

Réaliser en groupe un projet de communication : site web, communication réseaux sociaux, conception de supports de communication, promotion sur salons et forums online et offline.

Compétence à acquérir :

Familiariser les étudiants aux méthodes de communication dans le cadre d'un projet concret et leur apprendre les bases de la communication en entreprise (oral et écrit, online et offline).

Mode de contrôle des connaissances :

Notes sur livraison des projets

Culture générale en finance

ECTS : 1.5

Finance des marchés 1

ECTS : 3

Description du contenu de l'enseignement :

Le but de ce cours est de présenter les marchés financiers et de donner un aperçu des outils informatiques utilisés pour gérer les produits dérivés.

1. Bloc introductif : tour d'horizon culturel des marchés financiers, étude de l'hypothèse d'Absence d'Opportunité d'Arbitrage en toute généralité.
2. Rappels sur les variables aléatoires, moments. Simulation d'une v.a., Box-Muller. Méthode de Monte-Carlo.
3. Processus discrets (temps indexé par un entier), portefeuilles de couverture. Modèle binomial et valorisation par arbres.
4. Processus continus (temps indexé par un réel positif), formule et modèle de Black and Scholes, volatilité implicite.
5. Simulation du mouvement brownien, schéma d'Euler et formule d'Ito. Grecques.

Compétence à acquérir :

Etre capable de refaire les exercices et comprendre comment s'organise le pricing d'un produit dérivé, dans un cadre simple.

Finance des marchés 2

ECTS : 3

Description du contenu de l'enseignement :

Il s'agit d'implémenter sur Python des problèmes classiques en finance. En particulier:

Calibration et implémentation du modèle de Black et Scholes, calcul de la volatilité implicite.

Gestion de portefeuille: calcul de portefeuilles optimaux, frontière efficiente.

Sur-réplication de payoff Européens sans probabilité de risque neutre à partir d'un historique.

Calcul de mesures de risques (VAR, CVAR).

Compétence à acquérir :

Savoir comprendre et implémenter des modèles en finance.

Mode de contrôle des connaissances :

Examen terminal.

Finance entrepreneuriale

ECTS : 3

Description du contenu de l'enseignement :

Ce cours aborde les principaux thèmes de la finance d'entreprises. Il apporte un aperçu des mécanismes financiers de création de valeur. Au travers de la théorie et d'exemples réels, il s'articule en trois parties : outils de base, financement des sociétés et fusions-acquisitions. Les étudiants auront l'opportunité de présenter en groupe et à l'oral une société de leur choix (une liste de sociétés sera proposée par l'enseignant). Cette présentation mettra l'accent sur les aspects financiers des sociétés étudiées et permettra de valider les acquis.

Partie I - Outils financiers de l'entrepreneur :Eléments de stratégie : les cinq forces d'un marché (concurrents, clients, fournisseurs, produits de substitution et nouveaux entrants), notions générales de compétitivité, Le modèle économique des sociétés, Les sources du financement : la dette et le capital, Le business plan : paramètres et modélisation, Rappel des méthodes de valorisation des sociétés.

Partie II - Financement des sociétés aux différents stades de développement :Création : business angels et capital risque - attente des intervenants, sur quels critères financent-ils les start-up ?Développement : privateequity - objectifs et fonctionnement des fonds, caractéristiques des principales places financières (New York, San Francisco, Londres), Maturité : introduction en bourse - contraintes et opportunités d'une IPO.

Partie III - Introduction aux fusions et acquisitions :Pourquoi les sociétés fusionnent-elles : synergies, intérêts financiers, Les principaux acteurs internes (actionnaires, management) et externes (avocats, banques d'affaires, banques de financement, auditeurs, consultants), Enjeux de la fusion des sociétés : risques, culture, facteurs exogènes, Principales étapes d'une acquisition : identification d'une cible, approche des actionnaires, due diligence, valorisation, négociation, rôle des intervenants externes (avocats, banquiers, auditeurs, consultants), intégration post-fusion.

Partie IV - Cas d'une entreprise : création, développement, positionnement, écosystème, analyse financière de la société.

Gestion de portefeuille et risque

ECTS : 3

Description du contenu de l'enseignement :

Rendement/risque d'un portefeuille. Théorie moderne de gestion d'un portefeuille. Value atRisk.

Analyse de performance (caractéristiques d'un portefeuille).

Théorie moderne du portefeuille ; le modèle de Markowitz, CAPM.

Gestion/Allocation de portefeuilles et mesures de performance.

Modèles multifactoriels.

Mesures de risque : Value at Risk.

Gouvernance des systèmes d'informations bancaires

ECTS : 3

Les fondamentaux de la gestion d'actifs numériques

ECTS : 1.5

Livret d'entreprise et apprentissage

ECTS : 1

Machine Learning pour la finance

ECTS : 2

Management de Projets

ECTS : 3

Description du contenu de l'enseignement :

- Les différents processus qui accompagnent les méthodes de conduite de projet Systèmes d'informatique
- Rôle de la maîtrise d'ouvrage et de la maîtrise d'œuvre
- Les activités de support
- Exposer et développer les méthodologies de conduite de projet informatique
- Rétrospective méthodologique
- Définition et domaines d'usage d'un ERP
- Pourquoi un ERP ? Les enjeux Business
- Fonctionnement d'un ERP,
- Analyse et Stratégie de l'Entreprise
- Critères de choix d'un ERP

- Spécificités du projet BI

- Organisation du projet et facteurs clés du succès
- Développement de la solution, test de la solution, mise en production de la solution
- Évaluation de la solution

- la prévision des tendances IT

- Le rôle du Design Thinking en management de projets

Compétence à acquérir :

Ce cours vise à développer les compétences nécessaires en matière de lancement, planification, et gestion de projets de manière efficace ainsi qu'à la consolidation des notions de conception et de modélisation et de leur intégration dans une démarche agile.

Mémoire

ECTS : 8

Méthodes Agiles d'Ingénierie Logicielle

ECTS : 3

Description du contenu de l'enseignement :

La nouvelle économie met à rude épreuve les équipes des projets informatiques. L'instabilité notoire des spécifications et le raccourcissement des délais de livraison imposent une amélioration drastique de la réactivité et de la productivité du développement, sans tolérer pour autant une quelconque négligence en matière de stabilité et d'évolutivité des applications déployées. Les méthodologies dites "agiles" tentent de répondre à ces exigences, en adoptant un investissement minimaliste et prônant des résultats tangibles et fréquents, par opposition aux méthodologies classiques qui - tout en imposant des organisations coûteuses - n'arrivent plus à prouver leur efficacité dans ces conditions extrêmement dynamiques. Ce cours propose un panorama des méthodes et techniques efficaces applicables dès les projets de petite taille, y compris pour assurer la traçabilité complète entre les besoins et les composants techniques en continue évolution. Les étudiants pourront les mettre en œuvre avantageusement, preuve comprise, aussi bien pour leurs projets pédagogiques qu'industriels, une fois dans l'entreprise.

- Introduction. Evolution des modèles du cycle de vie des projets. Manifeste agile: valeurs et principes.
- Offre des méthodologies agiles : XP, Scrum, FDD, Kanban, Lean, etc.
- eXtremeProgramming : cycle de négociation, métaphore & storytelling, développement piloté par les tests, refactoring permanent, etc.
- Gestion des exigences et de configuration logicielles. CMMi vs agile.
- Zoom sur le testing automatique fonctionnel : Behaviour Driven Development en Gherkin avec Cucumber ou Behave.
- Modélisation, patterns agiles, stratégies et styles architecturaux agiles: Design Patterns, Analysis Patterns, Metapatterns, Hexagonal, CQRS, DDD.
- Techniques encourageant l'agilité : RAD, AOP, Assemblage de composants, MDA, métriques & audits, EventModeling, Impact et StoryMapping.

Illustrations en Java et Python, tout au long du cours.

Compétence à acquérir :

- Découvrir, comprendre et appliquer les principes et les pratiques agiles les plus efficaces.
- Collaborer en équipe pour atteindre rapidement un objectif fonctionnel ou technique.
- Renforcer les compétences de spécification, conception, planification, communication, développement et tests automatisés.
- Fournir des résultats tangibles et fréquents, avec le sourire

Mode de contrôle des connaissances :

Projet évolutif avec soutenance validant les connaissances individuelles acquises.

Bibliographie, lectures recommandées :

Agile Ownership in a nutshell (vidéo) — <https://youtu.be/502ILHjX9EE>

Scrum and XP from the Trenches — <https://leanagiletraining.com/wp-content/uploads/2020/03/Scrum-and-XP-from-the-Trenches-2nd-edition.pdf>

The Pragmatic Programmer — <https://pragprog.com/titles/tpp20/the-pragmatic-programmer-20th-anniversary-edition/>

Le blog de Martin Fowler — <https://martinfowler.com>

Le manifeste agile — <https://agilemanifesto.org>

Extreme programming: a gentle introduction — <http://www.extremeprogramming.org>

The Scrum Guide — <https://www.scrum.org/resources/scrum-guide>

Projet Programmation financière

ECTS : 1.5

Description du contenu de l'enseignement :

Ce cours projet traite du problème d'implémentation des modèles de valuation des produits dérivés financiers.

Il s'agira de mettre en application les modèles de pricing présentés en finance de marché. Ce cours-projet abordera tout particulièrement la méthode de Monte-Carlo et son implémentation en Python afin d'évaluer des options Européennes et Asiatiques dans des modèles à volatilité locale.

Compétence à acquérir :

Programmation en Python des méthodes de pricing vues dans le cours Finance de Marché.

Mode de contrôle des connaissances :

Contrôle sous la forme d'un TP.

SQL, NoSQL et NewSQL

ECTS : 3

Description du contenu de l'enseignement :

Le cours a pour objectif d'apprendre aux étudiants les aspects fondamentaux des différents types bases de données qu'elles soient basées sur le SQL, le NoSQL (Not Only SQL) ou récemment le NewSQL.

Le cours s'articule en trois parties.

Dans la première partie, l'accent est mis sur les bases de données relationnelles : leurs avantages et leurs inconvénients, ainsi que la correspondance objet-relationnel (Object-Relationnel Mapping -ORM) avec la norme JPA.

La deuxième partie présentera les différents modèles noSQL (clé-valeur, document, graphe), les notions de disponibilité et de partitionnement à la cohérence (propriétés BASE, théorème CAP), les différents systèmes NoSQL (Redis, MongoDB, Cassandra, Neo4j, ...), les avantages et les inconvénients du NoSQL.

La troisième partie sera consacrée aux bases NewSQL : leur définition et leurs caractéristiques, les nouvelles architectures et la notion de DBaaS (Database as a service), leurs avantages et leurs inconvénients.

Les notions apprises seront mises en pratique dans le cadre de TP (sur Redis, Cassandra, MongoDB et Neo4j) et d'un projet où les étudiants devront manipuler différents types de bases de données afin de les comparer.

Compétence à acquérir :

- Compréhension des modèles NoSQL
- Modélisation des bases de données en SQL et NoSQL
- Norme JPA.

Mode de contrôle des connaissances :

Cette UE est évaluée par un projet (50%) et un examen (50%)

Bibliographie, lectures recommandées :

P. J. Sadalage et M. Fowler (2013). *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Addison Wesley

R. Bruchez (2015). *Les bases de données NoSQL et le Big Data*. Eyrolles

G. Harrison (2015), *Next Generation Databases, NoSQL, NewSQL and Big Data*, Apress

Systems, Languages and Paradigms for Big Data

ECTS : 3

Description du contenu de l'enseignement :

Le cours a pour objectif d'apprendre aux étudiants les aspects fondamentaux des technologies Big Data pour la gestion et analyse de données massives.

Le cours s'articule en trois parties.

Dans la première, l'accent est sur le paradigme MapReduce et le système Hadoop, avec un focus sur son système de fichiers HDFS. Le cours illustrera les mécanismes de base de Hadoop pour le support de l'exécution parallèle de 'dataflow' MapReduce sur des clusters de machines. Une attention particulière sera donnée aux aspects algorithmiques et d'optimisation de dataflow MapReduce.

La deuxième partie présentera des langages de requête et d'analyse de données caractérisés par des mécanismes de haut niveau et qui sont compilé sur MapReduce. Le focus sera sur Hive, permettant du traitement de données via SQL. Des techniques de compilation de SQL vers MapReduce seront présentées.

La troisième partie sera consacrée à des évolutions de Hadoop, et en particulier au système Spark et au langage de support

Scala. Le focus sera sur l'architecture de Spark, la notion de RDD, l'évaluation lazy de transformations et actions sur des collections distribuées RDD.

Les notions apprises seront mises en pratique dans un projet où les étudiants devront concevoir un dataflow pour l'analyse de grands volumes de données. L'implémentation sera faite tant en MapReduce qu'en Spark, et une analyse expérimentale sera effectuée pour comparer les performances des deux implémentations.

Compétence à acquérir :

Savoir concevoir des applications pour le traitement efficace de données massives.

Mode de contrôle des connaissances :

CC et Projet

Systèmes distribués et nouvelles technologies Web en Java

ECTS : 3

Description du contenu de l'enseignement :

Le but est d'apporter de meilleures connaissances sur le développement de l'informatique et de comprendre comment la technologie favorise la transformation digitale des entreprises.

Ce cours introduit les principes des systèmes distribués (distributed system), forme les étudiants aux nouvelles technologies du Web en utilisant Java et ses frameworks, leur fait découvrir le framework Angular pour développer une application front-end et leur permet de jeter un coup d'œil sur le Cloud Public comme Microsoft Azure.

Le cours se termine par un projet applicatif, où les étudiants seront capables de concevoir et de développer une application de A à Z, puis de la déployer sur un serveur Cloud et de la rendre accessible sur Internet.

Compétence à acquérir :

- Découvrir les fondamentaux de distributed system et la conception de haut niveau (high-level design).
- Savoir comment concevoir les REST APIs et développer une application front-end web, fullstack développement
- Comprendre les enjeux du DevOps, Cloud
- Framework: Java Springboot, Angular

Mode de contrôle des connaissances :

- 50% contrôle continu
- 50% projet avec soutenance