



**Département d'informatique
et de mathématique**
Université du Québec à Chicoutimi

PLAN DE COURS

Conception et architecture des systèmes d'infonuagique
(8INF876)
Automne 2025
Groupe : 01

Professeur

1. Description du professeur

Le devoir de tout professeur est de s'améliorer constamment (enseignement, recherche et services à la collectivité), telle est ma devise depuis que je suis à l'UQAC et que j'ai accompli ma formation continue dans les différents domaines émergents de l'informatique, tels que les systèmes distribués (Cloud Computing CC). Je travaille dans le domaine de recherche de conception des applications intelligentes, basé sur le génie logiciel et les systèmes distribués intelligents (Cloud et Pervasive Computing, IoT, ville et maison intelligentes, systèmes de santé intelligents, conteneurs & virtualisation, etc.).

Durant mon travail à l'UQAC, j'ai toujours été à l'écoute des étudiants et empathique envers les problèmes rencontrés par ceux qui demandent plus d'attention. J'utilise une méthode constructive pour permettre aux étudiants de réfléchir, en adoptant les stratégies suivantes :

- L'apprentissage par exemple concret pour faciliter la construction du savoir-faire chez les étudiants et dissiper les éventuelles difficultés;
- Les lectures par les étudiants (chapitres, articles, rapport de recherche, pages Web) pour enrichir leurs connaissances dans le domaine en question;
- L'apprentissage par question/réponse;
- L'écoute des étudiants pour discuter de leurs questions et donner des réponses.

En même temps, j'utilise des pages Web pour fournir les divers documents (notes de cours, laboratoires, travaux dirigés et pratiques, examen, vidéos, etc.). Ces documents sont ma propre réalisation dans la majorité des cas. Ce matériel est disponible sur le site du cours (Moodle) de l'UQAC.

2. Identification du cours dans le programme (Situation du cours dans le programme)

Le cours est optionnel dans les divers programmes des cycles supérieurs du département d'informatique et de mathématique (DIM) de l'université. Aucun cours n'est préalable au cours 8INF876. Cependant, ce cours nécessite la connaissance des concepts de base d'un réseautique (TCP/IP, adresse IP, ...), programmation objet (Java, C/C++, Python, JS, GO, ...), base de données (SQL, MySQL, ...), Eclipse, NetBeans, Visual Studio Code, etc. De plus, il demande une capacité d'analyse rigoureuse et méthodique. Ce cours montre les concepts et les technologies importants et utilisés pour concevoir des architectures distribuées (Cloud, Fog, Edge, IoT, C/S, P2P, ...), des applications IoT intelligentes, etc. Ce cours permet aux étudiants d'avoir une expertise complète avec les autres cours de notre programme de l'UQAC.

3. Description du cours

Objectif général

Concevoir et déployer des applications distribuées et infonuagiques :

- Concevoir les architectures des applications distribuées et infonuagiques
- Déployer des applications en utilisant différents modèles de services, déploiement et techniques
- Se familiariser aux diverses méthodes de virtualisation
- Analyser les exigences des applications distribuées infonuagiques

Concepts fondamentaux (modèles, types et architectures) des systèmes distribués et d'infonuagique. Exigences des environnements modernes du Cloud (ouverture, mise à l'échelle, transparence, fiabilité, performance et interopérabilité). Architectures des applications infonuagiques, indépendamment des plateformes matérielles et logicielles : client-serveur, N-tiers, Peer-to-Peer, Clustering, Grid, Cloud, Edge/Fog, IoT, hybride. Types de communication (message, appel des procédures et méthodes, multicast, et événement). Middlewares et protocoles de communication interservices (REST; gRPC; MQTT). Éléments de Cloud (puissance de calcul, stockage, réseau, analytique). Modèles de services du Cloud (Infrastructure IaaS, plateforme PaaS, applications SaaS et fonctions FaaS). Modèles de déploiement du Cloud (On-premises, privé, publique, multi-cloud, poly-cloud et hybride). Virtualisation dans le Cloud (Machines virtuelles VMs, conteneur et serverless). Techniques DevOps et CI/CD pour l'automatisation et le déploiement d'applications cloud-native. Technologies du Cloud intégrant Docker, Docker Compose, Kubernetes, Terraform, et GitHub.

Description du cours (objectifs spécifiques)

Un système réparti (SR ou SD) est un *ensemble de ressources physiques et logiques* géographiquement dispersées et reliées par un réseau de communication qui leur permet d'échanger de l'information en vue de coordonner la réalisation d'une tâche commune. Les systèmes répartis se distinguent par le fait que le traitement de l'information, pour une application donnée, est réparti entre plusieurs ordinateurs et appareils physiques autonomes reliés entre eux par des voies de communication. Les principales caractéristiques de tels systèmes répartis sont : *ouverture, fiabilité, performance, disponibilité, mise à l'échelle (extensibilité), adaptabilité-élasticité (CC), etc.* Ce cours vise à comprendre ces caractéristiques et à familiariser l'étudiant à la recherche et au développement des applications distribuées dans l'environnement de l'Internet et Cloud computing. Plus spécifiquement,

8INF876 – Conception et architecture des systèmes d'infonuagique	Plan de Cours	Aut 2025	Page 3 de 18
--	---------------	----------	--------------

- **Compréhension fondamentale** : Acquérir une compréhension solide des notions de base des systèmes répartis et de l'infonuagique, et leur importance dans le contexte actuel : les processus, la communication, l'échange de données, la synchronisation, l'environnement des applications réparties et les architectures émergentes.
- **Architecture et Design** : Analyser en profondeur les diverses architectures d'application distribuée et infonuagique, les types de systèmes distribués (*High performance distributed computing*, *Distributed Information systems*, *Pervasive computing*), les architectures (client-serveur, n-tiers, Peer-to-Peer, Cloud, IoT, Edge et Fog Computing, etc.), et les modèles de communication (synchrone, asynchrone, persistant, etc.) des systèmes répartis. Développer une aptitude à concevoir et mettre en œuvre des applications réparties/Cloud robustes et performantes (introduction de la méthode ADD : Attribute-Driven Design).
- **Virtualisation et Informatique** : Appréhender le rôle et l'importance de la virtualisation dans les systèmes répartis, et comment les concepts d'infonuagique, tels que IaaS, PaaS SaaS, et FaaS, transforment le paysage technologique.
- **Développement d'applications** : Comprendre le développement d'applications modulaires orientées composants-Micro-services (tolérantes aux défaillances et optimisées pour la performance), cultiver la capacité de la répartition d'applications : présentations, traitements et données vers des applications Microservices, etc.
- **Plateformes** : Décrire les concepts, les techniques et les plateformes qui ont été développés pour faciliter les tâches de développement et d'implémentation des applications distribuées, RMI, Socket, Web Socket, URLConnexion, servlet, JSP, Java EE (EJB), Spring Boot, gRPC, CakePHP, Symphony, SOAP, REST, NodeJS, IoT, Cloud Computing, VM, conteneurs (Docker, Kubernetes, DevOps-CI/CD), Serverless, Docker Swarm, etc.
- **Analyse critique et rédaction scientifique** : adopter une approche critique envers les technologies, les protocoles et les articles scientifiques existants, tout en explorant les défis et solutions émergentes dans le domaine des systèmes répartis. L'étudiant sera également initié à la rédaction de rapports/articles scientifiques, émergeant ainsi la recherche académique et fournissant une plateforme pour présenter ses propres découvertes et analyses dans le domaine.
- **Présentation et résolution de problèmes** : Encourager les étudiants à détecter des défis des systèmes distribués et Cloud Computing, à préparer et à présenter des exposés pour montrer ces défis. Les étudiants sont invités à présenter les approches existantes pour résoudre tels défis et à proposer des solutions : résoudre des problèmes relatifs aux systèmes répartis: plateformes réparties et web, middleware (Java EE, Spring, .Net, REST (Java-RS), etc.), service web, mobilité, IoT, MQTT, CoAP, LORA, WIFI, WIFI-Direct, Lopy4, Cloud Computing (Docker, Kubernetes, DevOps), Cloud, Firestore et Firebase, Fog et Edge Computing, OpenFog, Blockchain, P2P, NodeJS, etc.
- **Occasion de la recherche et des technologies** : Offrir une vision globale liée à plusieurs domaines de recherche en informatique liés aux systèmes distribués et Cloud computing : invitation possible des chercheurs ou collègues pour présenter leurs recherches/technologies

...

4. Contenu du cours

Il est possible de modifier l'ordre des chapitres dans le calendrier afin de permettre aux étudiants d'avancer dans la partie pratique en parallèle avec la partie théorique du cours. Le cours permettra aux étudiants l'apprentissage des éléments suivants (voir l'annexe 1 pour plus de détails) :

Partie théorique :

- Introduction à un système réparti & types de Computing : problèmes des applications réparties;
- Types et architectures des systèmes distribués et du Cloud Computing
- Modèles de communication, Réseaux et protocoles de communication
- Architecture des applications Cloud Computing, ADD, DevOps (CI/CD) & Microservices
- Modèles de services et Modèles de déploiement
- Cloud Computing : Virtualisation & isolation applicative (conteneur) & Serverless
- Architectures émergentes : Cloud & Fog computing, IoT, Peer-to-Peer

Partie Pratique :

- Communication par message, ex. Sockets, MOM, MQ, MPI
- Communication par requête-réponse : RPC, objet (RMI, Java EE), gRPC, REST
- Applications & Plateformes distribuées Web et Services web standards, REST, Ajax, SOAP, NodeJS
- Plateformes et Processus du Cloud Computing, ex : Google, Amazon, Microsoft (Azure), etc.
- Cadres et outils du Cloud : VM, haute disponibilité, Docker, Kubernetes, Docker Swarm, etc.
- Gestion et automatisation cycle de vie des applications/infrastructures Cloud : Ansible, Terraform
- Protocoles de communication : HTTP, CoAP, MQTT

5. ACTIVITÉS D'APPRENTISSAGE (Modalité d'évaluation)

a) Présentation et Critique d'un(e) article/protocole/technologie

Cette partie présente une critique d'un(e) article/protocole/technologie traitant un sujet des systèmes distribués. Certains liens vers des articles/protocoles/technologies sont disponibles sur le

8INF876 – Conception et architecture des systèmes d'infonuagique	Plan de Cours	Aut 2025	Page 5 de 18
--	---------------	----------	--------------

site du cours. Il est aussi possible d'aller chercher ces éléments dans des revues scientifiques (IEEE, ACM, Springer, Inderscience, MDPI, ou autre). La critique consiste en un exposé oral qui a pour but d'expliquer le problème posé, sa réalisation et les critiques que l'on peut y faire.

Activité	Critères
Présentation orale d'article/protocole/technologie (I)	• Clarté & Concision • Réponse aux questions • Compréhension du domaine d'application • Compréhension du problème résolu • Critique appropriée (avantages et inconvénients) • Références utilisées

Cette critique est réalisée individuellement ou en groupe des étudiants (à décider en classe). Le choix de cet article/protocole/technologie doit auparavant être validé par le professeur.

Quoi remettre :

- **Seulement l'exposé oral (ppt)** doit être **remis** au professeur (sur Moodle directement) pour montrer un résumé et une critique de l'article/protocole/technologie.

b) Projet du trimestre (2 ou 3 étudiants)

Le projet consiste à résoudre un problème relatif aux systèmes répartis. Ce projet comporte une partie d'analyse qui décrit comment le problème est résolu. La deuxième partie consiste à appliquer le problème en utilisant une technologie donnée de systèmes distribués. Le résultat de ce projet consiste à écrire un article.

Exemples de défis : architectures, méthodes de communication, analyses des données de IoT, etc.

- Informatique pervasive (ambiante) : Maison intelligente, ville intelligente, télésurveillance, systèmes de santé, IoT-M2M/Eclipse, MQTT, CoAP, Lora, Wifi-Direct, BLE, etc.
- Étude de cas/frameworks (implémentation, avantages et inconvénients): OpenStack, VMWare, Docker, Kubernetes, Matter IoT, Kafka, Bootstrap, REACT, Angular JS, NodeJS, Java EE, SpringBoot, etc.
- Algorithmes d'IA : Federated Learning, algorithmes de chemin le plus court, algorithmes d'IA interprétables (LIME, RETE, SHAP, etc.), algorithmes P2P, multicast, Clustering, Blockchain, algorithmes de cache dans les systèmes distribués, etc.

Deux rencontres avec les étudiants permettront de préciser les étapes :

- Avant le mi-trimestre : présentation (II) du problème à résoudre, des attendus du projet, et de la méthode de résolution
- Fin du trimestre: présentation du projet et des résultats obtenus (III).

Activité	Critères
Présentation orale II (ou une page décrivant le projet) (5 mns)	Clarté, Concision Définition du problème et comment vous allez le résoudre Réponse aux questions
Présentation orale III	Clarté, Concision

Des exemples de problèmes à résoudre seront disponibles sur le site du cours et discutés en classe.

Des travaux pratiques (deux ou trois travaux) doivent être réalisés en utilisant les diverses technologies de systèmes répartis. La date précise et les règles de présentation des travaux seront décrites lors de la remise des différents travaux à réaliser. ***L'étudiant est responsable de travailler avec les technologies utilisées dans ce cours.*** La description et la date de l'examen (fiche) seront détaillées en classe.

8INF876 – Conception et architecture des systèmes d'infonuagique	Plan de Cours	Aut 2025	Page 7 de 18
--	---------------	----------	--------------

Implémentation et/ou étude de cas ... À remettre : 15^{ème} séance (dernière semaine)	Présentation des résultats, Perspectives	
Travaux Pratiques	- Réalisation et Qualité du travail - Date de remise : seront spécifiées dans chaque TP	30%
Examen du cours	Date de l'examen : voir calendrier (13^{ème} séance)	35%

En cas d'échec dans ce cours, il n'y a pas une autre évaluation tenant lieu de l'activité. Cependant, une attestation d'un médecin en bonne et due forme, présentée au plus tard deux semaines après la date de l'évaluation et confirmant que l'étudiant était dans l'impossibilité de se présenter à une activité (présentation, article, rapport, etc.) pour des raisons de santé, pourra être considérée comme une justification d'absence valable.

Des consignes supplémentaires ou des modifications pourront être communiquées au cours du trimestre. Il est possible de remplacer l'examen par un travail pratique ou une fiche (Questions).

6. FORMULES PÉDAGOGIQUES ET ACTIVITÉS D'ENSEIGNEMENT

Les différentes séances prennent la forme d'exposés théoriques illustrés par des exemples spécifiés. L'étudiante, l'étudiant, est responsable d'effectuer préalablement les lectures correspondant au sujet de la semaine. L'étudiant(e) est le seul responsable de sa formation: Il étudie les concepts et les défis des systèmes repartis vus dans les séances et il fait plus de recherches supplémentaires à la maison.

Durant ce trimestre, je donne les séances en mode hybride (avoir la souplesse pour permettre aux étudiants de suivre les cours). Bref, les formules pédagogiques suivantes seront utilisées:

- L'enseignement est dispensé sous forme mixte (magistrale, interactive et asynchrone) avec l'utilisation des notes de cours, exemples, livres recommandés, chapitres à lire, sites Web à consulter, etc.
- Des démonstrations viennent illustrer certaines notions théoriques.
- Des recherches et des présentations d'articles pourront enrichir le cours et aider à mieux comprendre les défis actuels du domaine et leurs solutions.
- Des projets de recherche, une fiche (examen) et travaux pratiques se rapportant directement aux exposés compléteront l'apprentissage.

La formule pédagogique du cours utilise les modes asynchrone, synchrone et bimodale:

Le **mode asynchrone** permet aux étudiants de réaliser les démarches d'apprentissage qui leur sont proposées aux moments qui leur conviennent, que ce soit pour consulter les ressources d'apprentissage mises à leur disposition ou pour interagir avec le professeur ou les autres étudiants. Quant à la **modalité synchrone**, elle fait référence à des séances offertes en prestation virtuelle à des moments déterminés en faisant appel à la vidéoconférence. On peut aussi utiliser la mode bimodale pour permettre aux étudiants plus de souplesse. Les **modalités** peuvent aussi se conjuguer dans une même séance ou varier d'une séance à l'autre au cours du trimestre (mixte).¹

7. MODALITÉS DE COMMUNICATION ET D'ENCADREMENT

Périodes de disponibilité

- Mardi de 11h00 à 12h00 sur Zoom (voir le lien de consultation), ou
- Rendez-vous par adresse électronique
- Rencontre virtuelle quand nécessaire : Zoom (voir le lien sur Moodle) ou Teams, ou Google Meet

Le professeur se rendra disponible sur ces outils (Zoom, Teams ou Google Meet) en dehors des heures régulières du cours les mardis. Veuillez noter qu'il est possible que le professeur soit également disponible en dehors de cette période. Vous pouvez prendre rendez-vous via courriel auprès du professeur par courriel:

Il devient important que nous utilisions des différentes modalités d'encadrement (échange par courriel, rencontre d'encadrement de groupe ou de façon individuelle, avec ou sans rendez-vous, etc.).

8. DATES IMPORTANTES

Dates importantes	
Date de début du cours	25 août 2025
Semaine de relâche	14-17 octobre 2025

¹ Basque, J., & Baillargeon, M. (2013). La conception de cours à distance. Le Tableau, 2(1). [en ligne] <http://pedagogie.quebec.ca/sites/default/files/documents/numeros-tableau/letableau-v2-n1-2013.pdf>

Date limite de modifications et d'annulations de cours avec remboursement ou retrait de la charge financière	08 septembre 2025
Date de fin du cours	15 décembre 2025

9. CALENDRIER APPROXIMATIF DU COURS

Sem. / Séance	Contenu : calendrier approximatif des activités (L'ordre et le contenu peuvent être modifiés selon les besoins des activités du cours)
1	Présentation du cours : • Objet et portée du cours • Documents de référence • Préalables • Présentation du contenu • Organisation des activités
2	Introduction : Systèmes distribués et Cloud computing • Qu'est-ce qu'un système distribué (SD) et qu'est-ce que le Cloud computing CC? • Caractéristiques des SD (<i>défis</i>). • Exemples et applications. • Générations des SD, Middleware. • Paradigmes informatiques
3	Types et architectures des SDs et Cloud : • Conception des applications réparties. • Types et architectures de systèmes distribués. • Modèles : Clients, Serveurs, Peer-to-Peer, Cloud, Micro-services. • Types de services. • Introduction à la méthode ADD (Attribute-Driven Design)
4	Modèles et protocoles de communication

	- Types de communication : persistante, discrète, synchrone, etc. - Rappels sur les protocoles de communication et la pile TCP/IP. - Processus, Ports, adresse IP, etc. - Communication par message : - Communication entre <i>processus dans l'environnement repart</i>, <i>Concept d'un socket</i> - Étude de cas : la communication par sockets (message) en Java, Python.
5	Modèle de communication à distance par requête-réponse (procédure, objet) : RPC, RMI, gRPC - Moyens de communication entre processus distribués par RPC ou gRPC ou RMI (communication par procédure) et les IDL, etc. - Moyens de communication entre processus distribués par méthode. Interfaces applicatives (API) : Les RMI synchrones et asynchrones. - gRPC, Java EE, Spring.
6	Cloud computing - Éléments du Cloud computing. - Modèles de déploiement. - Modèles de services. - Exemples.
7	Applications et plateformes distribuées et le Web - Applications distribuées. Le Web, CGI en PHP, Servlets - Communication asynchrone. Ajax: Objet XMLHttpRequest, Bootstrap - Service Web. REST, SOAP, XML, WSDL et WDDI, - Plateformes web, ex. NodeJS, Symfony, etc. - Web server Nginx
8	Cloud Computing : Virtualisation & isolation applicative - Virtualisation, VM, Étude de cas : VirtualBox. - Conteneur : isolation applicative. - Exemple de Conteneur : Docker. - MV vs Conteneurisation.
9	Cloud Computing : conception et architecture des systèmes CC - Architecture de Docker, & Kubernetes. - Cycle de vie de développement Docker de bout en bout.

	Exemples.
10	Cloud Computing : développement et déploiement. - DevOps : introduction, développement et déploiement pipeline. - Cycle de vie de CI/CD, Micro-services.
11	Introduction à l'informatique ubiquitaire et de l'IoT - IoT : Introduction et concepts. - Architectures IoT. - Exemple : CoAP, MQTT, Lora, Wifi-Direct, etc.
12	Architectures émergentes de Cloud Computing & Peer-to-Peer - Cloud & pervasive Computing, Fog & Edge Computing, P2P - Concept, couches, architecture, exemples, avantages et désavantages. - Exemples
13	Examen du cours : 13 ^{ième} séance
14	Correction de l'examen Présentation finale, Kafka, exemples
15	Présentation finale du projet final (ppt). Remettre les documents des projets finaux (présentation finale, rapport et code)

10. SYSTÈME DE NOTATION

La note de passage est fixée à 60 % ou C.

Il est indispensable de noter que la moyenne de la classe est la valeur clé pour calculer les notes finales des étudiants. Donc, selon la moyenne de la classe, les cotes seront attribuées aux étudiants.

11. DROIT D'AUTEUR ET PROPRIÉTÉ INTELLECTUELLE DU MATÉRIEL

L'ensemble du contenu rendu disponible dans le cadre de ce cours est protégé par la [Loi sur le droit d'auteur](#) et la Politique relative aux droits d'auteur de l'UQAC. Il est destiné à l'usage des étudiants dans le cadre de leur formation universitaire et ne doit pas être utilisé pour d'autres fins. Pour ce faire, les étudiants peuvent le consulter et, lorsque disponible au téléchargement, en faire une copie uniquement pour leur usage personnel.

Le contenu ne peut autrement être téléchargé ou copié, il ne peut pas non plus être reproduit, publié à nouveau, affiché, transmis, communiqué, diffusé, distribué, adapté ou modifié par quelque moyen

De plus, comme mentionné dans le manuel de gestion de l'UQAC, toute communication ou diffusion par l'entremise des TIC comme les ordinateurs portables, les cellulaires, les téléphones intelligents, les tablettes numériques, etc., ne peut être faite sans la permission de l'enseignant (https://www.uqac.ca/direction_services/secretariat_general/manuel/3/083.pdf).

Ainsi, l'enregistrement sonore ou visuel d'une activité synchrone par les étudiants est strictement interdit à moins d'obtenir au préalable une autorisation écrite du professeur.

12. SERVICES OFFERTS À L'UQAC

Différentes ressources pourraient venir en aide aux étudiants dans le cadre de leur formation (Centre de référence le Cube, le Service aux étudiants, les Services aux étudiants en situation de handicap, etc.). Voici brièvement ces différentes ressources (lien direct vers le site internet de ces ressources) :

<http://sae.uqac.ca/>

Il est aussi possible de communiquer avec la bibliothèque pour emprunter des livres et des articles.

13. POLITIQUE CONTRE LE HARCÈLEMENT ET LA VIOLENCE

Principes généraux

- L'Université du Québec à Chicoutimi s'est dotée d'une politique afin de prévenir et de faire cesser le harcèlement et la violence ;
- L'université favorise un milieu propice à la dignité et à l'estime de tous les membres de la communauté ;
- L'Université désavoue toute forme de harcèlement et de violence parce qu'ils constituent une atteinte aux droits de la personne ;

- Tous les membres de la communauté universitaire ont droit à un milieu de travail et d'études exempt de harcèlement et de violence.

Application de la politique

- Toute forme de harcèlement et de violence est prohibée;
- Tout membre de la communauté universitaire a le devoir de s'approprier la présente politique et de saisir le Comité institutionnel contre le harcèlement et la violence, le responsable du Bureau de réception des plaintes ou les personnes-ressources de toute situation qu'il juge en contravention de la présente politique.

Pour en savoir plus, nous vous recommandons les sites suivants :

http://www.uqac.ca/direction_services/secretariat_general/manuel/5/150.pdf

<http://www.uqac.ca/harcelement/index.php>

14. EXIGENCES DE FORMATION

Ce cours sera évalué en fonction de la Procédure relative à l'évaluation des activités aux programmes d'études de cycles supérieurs. Comme il en est fait mention dans le Manuel de gestion à l'adresse suivante :

http://www.uqac.ca/direction_services/secretariat_general/manuel/index.pdf (3.1.2-008).

Tout travail remis doit être conforme aux exigences de la politique institutionnelle en matière de maîtrise du français écrit. Comme il en est fait mention dans le Manuel de gestion à l'adresse suivante :

http://www.uqac.ca/direction_services/secretariat_general/manuel/index.pdf (3.1.1-012).

Tout travail dont la qualité du français serait jugée non conforme par le professeur, pourra être pénalisé jusqu'à concurrence de 10% du résultat maximal prévu.

- Les travaux déposés à l'enseignant pourront, en cas de doute, être soumis au logiciel pour une analyse. Pour éviter le plagiat dans vos travaux ou dans vos examens, nous vous suggérons de consulter votre bibliothécaire.

Conformément à la politique institutionnelle, tout acte de plagiat ou de fraude dans un cours peut entraîner l'annulation d'un élément d'évaluation ou l'échec du cours. Les actes de

plagiat ou de fraude peuvent même entraîner la suspension ou l'exclusion de l'étudiant de l'UQAC (CAD-8462). **Sont considérés comme plagiat, les actes suivants : obtenir ou tenter d'obtenir les réponses de quelqu'un pendant un examen ou participer à un échange de réponses;** utiliser totalement ou en partie un texte d'autrui en le faisant passer pour sien ou sans en indiquer les références; participer, tenter de participer à une substitution de personnes lors d'un examen ou d'un travail faisant l'objet d'une évaluation; posséder ou utiliser pendant un examen tout document ou matériel non autorisé.

Voici la définition de plagiat que l'on retrouve dans la procédure :

Utiliser totalement ou en partie le texte d'autrui ou tout matériel obtenu par quelque mode que ce soit, notamment par l'utilisation de ressources informatiques, tel Internet, en le faisant passer pour sien ou sans en indiquer les références.

- **Règles concernant les retards dans la remise des travaux.** Lorsqu'un travail est remis après la date d'échéance, l'étudiant(e) perd 10% pour chaque jour de retard. Les jours de fin de semaine sont comptés comme les jours de semaine. Après 7 jours, le travail/examen ne peut plus être remis et la note est 0.

15. Références

1. Notes du cours 8INF876 (obligatoires).
2. Chellammal Surianarayanan, Pethuru Raj Chelliah, Essentials of Cloud Computing, A Holistic Perspective, Springer, 2019.
<https://link.springer.com/book/10.1007/978-3-030-13134-0>
3. Brian Zambrano. Serverless Design patterns and Best Practices. Packt Publishing, Birmingham - Mumbai, 2018.
4. Tod Golding. Building Multi-tenant SaaS Architectures: Principles, Practices and Patterns using AWS, O'Reilly, 2024.
5. Sam Newman, Building Microservices: Designing Fine-Grained Systems, O'Reilly, 2021.
6. Steen, M.V., & Tanenbaum, A.S. (2017). *Distributed systems* (3e éd.). USA : CreateSpace Independent Publishing Platform.
<https://uqac.on.worldcat.org/oclc/1107759131>
7. Coulouris, G.F., Dollimore, J., Kindberg, T., & Blair, G. (2012). *Distributed systems: concepts and design* (5e éd.). Boston, MA: Addison-Wesley.
<https://uqac.on.worldcat.org/oclc/746622042>
8. Comer, D. (2015). *Computer networks and Internets* (6e éd.). Upper Saddle River, NJ: Pearson/Prentice Hall.
<https://uqac.on.worldcat.org/oclc/898164631>
9. Mark Richards & Neal Ford. Fundamentals of Software Architecture, an Engineering Approach, O'reilly, 2020. <https://www.oreilly.com/library/view/fundamentals-of-software/9781492043447/>
10. Dordogne, J. (2020). *Réseaux informatiques : notions fondamentales* (8e éd.). St-Herblain, France: Éditions ENI.
<https://uqac.on.worldcat.org/v2/oclc/1002203902>
11. Defrance, J.-M. (2013). *jQuery-Ajax avec PHP: 44 ateliers pour maîtriser jQuery* (4e éd.). Paris, France: Eyrolles.
<https://uqac.on.worldcat.org/v2/oclc/879377222>

-
12. Emmerich, W. (2000). *Engineering distributed objects*. Chichester, Royaume-Uni: Wiley.
<https://uqac.on.worldcat.org/v2/oclc/181838194>
 13. Fron, A. (2015). *Architectures réparties en Java : Middleware Java, services web, messagerie instantanée, transfert de données*. Paris, France: Dunod.
<https://uqac.on.worldcat.org/v2/oclc/1104268696>
 14. Gray, J.S. (2003). *Interprocess communications in Linux: The nooks and crannies*. Upper Saddle River, NJ: Prentice-Hall.
 15. Harold, E.R. (2014). *Java network programming* (4e éd.). Sebastopol, CA: O'Reilly Media.
<https://uqac.on.worldcat.org/v2/oclc/890439718>
 16. Horstmann, C.S. (2019). *Core Java (2 volumes)* (11e éd.). Boston, MA: Pearson.
<https://uqac.on.worldcat.org/v2/oclc/1088806332>
 17. Howser, G. (2020). *Computer networks and the Internet* (1ere éd.). Cham, Suisse: Springer.
<https://uqac.on.worldcat.org/v2/oclc/1135203808>
 18. Josh, J. (2018). *Java EE 8 recipes : A problem-solution approach* (2e éd.). New York, NY: Apress.
<https://uqac.on.worldcat.org/v2/oclc/1040031812>
 19. Lacroix, J. (2015). *Mastering Linux network administration*. Birmingham, Royaume-Uni: Packt.
<https://uqac.on.worldcat.org/v2/oclc/930602035>
 20. Michael, M. (2018). *Practical JSF in Java EE 8 : Web applications in Java for the enterprise* (1ere éd.). New York, NY: Apress
<https://uqac.on.worldcat.org/v2/oclc/1038471495>
 21. Murach, J., & Urban, M. (2014). *Murach's java servlets and JSP* (3e éd.). Fresno, CA: Mike Murach & Associates.
 22. Pierre, S. (2011). *Réseaux et systèmes informatiques mobiles : fondements, architectures et applications*. Montréal, QC: Presses internationales Polytechnique.
<https://uqac.on.worldcat.org/v2/oclc/794300589>
 23. Stevens, W.R. (2009). *UNIX Network Programming, Volume 2: Interprocess Communications* (2e éd.). Englewood Cliffs, NJ: Prentice-Hall.

<https://uqac.on.worldcat.org/v2/oclc/898164693>

24. RabbitMQ tutorials : <https://www.rabbitmq.com/getstarted.html>

25. Liste de sites intéressants sur mon site web :

<http://www.uqac.ca/portfolio/hamidmcheick/communications/>

26. Notes de cours des professeurs : Barrak, Taher, ...

27. Source de plan : adapté de <https://jenseigneadistance.telug.ca/>