

2024/2025

Livret pédagogique

**FILIÈRE INITIALE SOUS
STATUT APPRENTI (FISA)**

Campus de Poitiers

SOMMAIRE

ORGANISATION PEDAGOGIQUE.....	2
-------------------------------	---

MASSES HORAIRES ET DESCRIPTION DES MODULES

PREMIÈRE ANNÉE **7**

SEMESTRE 5 **7**

ÉLECTRONIQUE 1	7
PROGRAMMER AVEC JAVA – S5	8
AUTOMATIQUE ET TRAITEMENT DU SIGNAL.....	9
OUTILS SCIENTIFIQUES POUR L'INGENIEUR	9
LANGUES ET COMMUNICATION	13

SEMESTRE 6 **15**

COMMUNICATION LIFI.....	15
PROGRAMMER AVEC JAVA – S6	15
CONVERSION DE L'ÉNERGIE ÉLECTRIQUE.....	16
SYSTÈMES NUMÉRIQUES	17
LANGUES ET CULTURE GÉNÉRALE	19
MODULES ÉLECTIFS DE CULTURE GÉNÉRALE.....	21

DEUXIÈME ANNÉE **23**

SEMESTRE 7 **23**

ÉLECTRONIQUE	23
INFORMATIQUE	23
SYSTÈMES EMBARQUÉS.....	25
LANGUES ET GESTION	26
PARCOURS INFORMATIQUE.....	28

SEMESTRE 8 **31**

ENSEIGNEMENTS TRANVERSAUX	31
---------------------------------	----

SEMESTRE 8 – DOMINANTE DÉVELOPPEMENT LOGICIEL TEST & QUALITÉ **34**

ANALYSE ET CONCEPTION	34
DÉVELOPPEMENT LOGICIEL, ARCHITECTURE & TEST	37

SEMESTRE 8 – DOMINANTE INTELLIGENCE ARTIFICIELLE & BIGDATA **41**

MATHÉMATIQUES POUR L'ANALYSE DE DONNÉES	41
INTELLIGENCE ARTIFICIELLE	43

BIG DATA.....	45
TROISIÈME ANNÉE	48
SEMESTRE 9 – DOMINANTE DÉVELOPPEMENT LOGICIEL - TEST & QUALITÉ	50
QUALITÉ LOGICIELLE	50
ARCHITECTURE ET SÉCURITÉ	52
CLOUD & DEVOPS	54
PROJET D'INTÉGRATION	56
SEMESTRE 9 – DOMINANTE INTELLIGENCE ARTIFICIELLE & BIGDATA	58
INTELLIGENCE ARTIFICIELLE ET BIG DATA.....	58
DATA GOUVERNANCE, GREEN IA ET APPROCHE RESPONSABLE DE L'IA	62
PROJET IA & BIG DATA	64

1^{ère} Année (Semestres 5 et 6)

La première année, d'un volume de 650 heures en moyenne, constitue majoritairement un enseignement de base commun à tous les élèves ingénieurs.

Le tronc commun généraliste, qui suit le parcours de découverte, fournit les connaissances scientifiques et les compétences managériales et linguistiques indispensables à un ingénieur généraliste. Il représente environ 85% du volume horaire total de première année.

Ces enseignements de tronc commun portent sur :

- L'approfondissement des connaissances scientifiques de base (Mathématiques, Physique) pour environ 13,2% du volume horaire total de première année,
- L'acquisition de compétences techniques propres à nos domaines (télécommunications, informatique et réseaux, électronique, systèmes embarqués, électricité industrielle, automatique) pour environ 69% du volume horaire total de première année,
- L'approche, le perfectionnement des sciences humaines utiles à l'ingénieur (communication, culture générale, gestion financière, langues) pour environ 17% du volume horaire total de première année.

2^{ème} Année (Semestres 7 et 8)

La seconde année, d'un volume de 509 heures en moyenne, comprend deux parties :

- des enseignements de tronc commun ainsi que des parcours technologiques représentant environ 63% du volume horaire total. Pour les apprentis, le parcours se résume au choix d'un département en fonction des missions en entreprise.
- des enseignements de dominante représentant environ 36,2% du volume horaire total.

Les enseignements de tronc commun sont répartis sur les deux semestres qui constituent l'année.

Ils portent sur :

- la poursuite de l'acquisition des compétences techniques dans nos domaines (télécommunications, informatique et réseaux, électronique, systèmes embarqués, électricité industrielle, automatique),
- l'acquisition de compétences humaines, managériales et linguistiques,

Les enseignements de dominante débutent au second semestre.

Ces dominantes, qui offrent au total 2 parcours possibles aux étudiants, sont :

- Développement Logiciel - Test & Qualité (bilingue)
- Intelligence Artificielle & BigData (bilingue)

La troisième année se décompose en trois parties :

- une partie commune à tous les étudiants,
- des enseignements spécifiques à chaque dominante,
- la fin de la mission en entreprise.

Les deux premières parties représentent environ 450 heures d'enseignements et se déroulent au premier semestre.

La partie commune à tous les étudiants représente environ 50% du volume total horaire et comporte :

- des enseignements managériaux et une sensibilisation aux métiers par le choix d'une approche métier parmi les 10 proposées :

Ingénieur Conseil

Ingénieur Recherche et Développement

- le projet ingénieur.

Les enseignements spécifiques de dominantes représentent environ 50% du volume total horaire et comportent :

- Des enseignements technologiques obligatoires pour chaque dominante

MASSES HORAIRES, COEFFICIENTS ET
CONTROLES
1ère année - Semestres 5 et 6
FILIERE INITIALE STATUT APPRENTI
Année 2024/2025

UE Semestre 5	Réf.	Module	SEM.	COURS	TD	TP	TUT	APP	EXAMEN	TOTAL	COEF	ECTS	CONTROLES
ELECTRONIQUE 1	E1C1-F	Electronique 1	5	7	16	28		4	4	59	8	3	2 CE (25%/25%) /CITP (25%) /CCTP (25%)
	PJ1C1-F	Programmer avec java - S5	5			0	15	24	2	41	8	3	2 CE (40%, 50%) , 1 CC (10%)
AUTOMATIQUE ET TRAITEMENT DU SIGNAL	ATSC1-F	Automatique et Traitement du signal	5	11	16	27			4	58	11	3	2 CE (50%, 50%)
	OS1C1-F	Electromagnétisme (20%)	5	10	10				2	22			CE
	OS1C2-F	Matths (40%)	5	10	16				2	28			CE
	OS1C3-F	Probabilités (20%)	5	10	10				2	22	9	3	CE
	OS1C4-F	Thermique (20%)	5	10	6				2	18			CE
LANGUES ET COMMUNICATION	LC1C1-F	Anglais (70%)	5		28					28	11	3	CC1 écrit (20%), CC2 oral (20%), CC3 écrit (30%), CC4 écrit (10%), CC5 Production/Participation (20%)
	LC1C2-F	Communication (30%)	5	14					2	16			CO (30%) + CE (50%) + Projet Voltaire (20%)
TRANSITION ECOLOGIQUE		Transition écologique	5	11	4	3			2	20	3	2	1 CE (75%), SULTEST (25%)
		TRAVAIL EN AUTONOMIE								20			
UE Semestre 6		Module	SEM.	COURS	TD	TP	TUT	APP	EXAMEN	TOTAL	COEF	ECTS	CONTROLES
COMMUNICATION LIFI	CL1C1-F	Communication LIFI	6	8		40			2	50	9	3	CCTP (25%), CE (25%), CCTP (25%), CE(25%)
	PJ1C1-F	Programmer avec java - S6	6			28	8	16	2	54	9	3	CE: 1h (40%), CE : 1h (40%) CC de projet (20%)
CONVERSION DE L'ENERGIE ELECTRIQUE	CEE1C1-F	Conversion de l'énergie électrique - EK	6	12	12				4	36	12	3	CE (30%)
	CEE1C2-F	Conversion de l'énergie électrique - EP		8									CE (30%)
	CEE1C3-F	Conversion de l'énergie électrique - TP	6			15				15			CITP (40%)
SYSTEMES NUMERIQUES	SN1C1-F	Logique	6	8	12	12			2	34	5	3	CE (50%)/CCTP (50%)
	SN1C2-F	Développement en électronique embarquée	6		4	8			1	13	2		1 CE
LANGUES ET CULTURE GENERALE	LOG1C1-A	Anglais 60%	6		44,5					44,5	13	4	CC1 Mini-projet (50%), CC2 TOEIC Blanc (30%), CC3 Production/Participation (20%)
	LOG1C2	Electifs de culture Générale* 20%	6	20						20			CC
	LOG1C3-F	Gestion Financière 20%	6	16					2	18			CE
ENTREPRISE		Entreprise	5 & 6									27	Rapport/Soutenance + Evaluation pratique par l'Entreprise
		TRAVAIL EN AUTONOMIE								35			
				155	178,5	158	26	44		651,5	100	60	

* MODULES ELECTIFS Culture générale (1 Module à choisir)										TOTAL
UE	Réf.	Module	SEM.	COURS						
Langues et Culture Générale	LOG1E3-F	Développement Durable : Numérique Responsable	6	20						20

CC : Contrôle Continu / CE : Contrôle Ecrit / CO : Contrôle Oral / CCTP : Contrôle Continu de TP / CITP : Contrôle Individuel de TP

SEMESTRE 5

ÉLECTRONIQUE 1

OBJECTIFS D'APPRENTISSAGE

Savoir paramétrer, simuler une fonction électronique de base et choisir les appareils de mesures pour tester le fonctionnement du montage

Objectifs fondamentaux :

- Savoir choisir des appareils de mesures, mettre en œuvre un banc de tests (typiquement 2 sources et 3 appareils de mesure) et l'exploiter,
- Savoir utiliser (câbler, régler, lire un résultat) un oscilloscope, un multimètre, un GBF, une alim DC
- Savoir calculer les tensions et courants et déterminer les grandeurs caractéristiques (gain, impédances, ...) dans un schéma comprenant au plus un élément actif et quelques branches (3 à 4),
- Savoir dessiner et traduire un schéma électrique en câblage
- Être capable d'évaluer la validité d'un résultat (homogénéité des équations, ordre de grandeur).
- Savoir lire et exploiter une fiche technique.

PROGRAMME

Les théorèmes de circuits électriques :

- Les théorèmes Kirschhoff (nœuds, maille, diviseur), Millman, et en reconnaître l'usage.
- Ecrire et manipuler les impédances complexes (R, C, L).
- Calculer la fonction de transfert d'une fonction électronique de base, tracer son diagramme de Bode

Les montages de base des amplificateurs opérationnels (AO)

- Caractéristiques principales d'un AO
- Montages de base en régime linéaire (inverseur, non inverseur, Additionneur, soustracteur, instrumentation, logarithmique, exponentiel,)
- Impact des défauts de l'ampli op réel sur les montages de base

Les filtres du 1^{er} du 2^{ème} ordre

- Notion de filtrage et diagramme de Bode
- Les différents filtres (P.Bas, P.Haut, P.Bande, Coupe bande)
- Synthèse de filtres à structure de Rauch ou Sallen-Key

Les appareils de mesures usuels et leurs utilisations

- Traduire un schéma en câblage (en utilisant correctement les cordons BNC et 4mm).
- Mesurer une tension avec un oscilloscope, un multimètre.

- Générer une forme d'onde, la reconnaître à l'oscilloscope, en régler les paramètres.
- Mesurer une réponse en fréquence.
- Mettre en forme et interpréter les résultats.
- Utiliser Excel pour tracer une courbe.

Le logiciel de CAO ORCADE PSPICE :

- Transcrire un problème en schéma de simulation.
- Choix du mode de simulation approprié (Bias, DC, Transient, AC)
- Utiliser le probe pour tracer les courbes appropriées

PROGRAMMER AVEC JAVA – S5

OBJECTIF

A partir d'un énoncé pouvant inclure un diagramme de classes UML (2 classes liées par une association), écrire, tester et documenter une application Java.

OBJECTIFS D'APPRENTISSAGE

- A partir d'un énoncé, choisir les structures de contrôles, déterminer les variables, et la séquence d'instruction permettant de produire le résultat attendu (à l'écrit)
- Utiliser un outil de débogage sur une méthode (point d'arrêt, visualiser contenu d'une variable et son évolution)
- Caractériser un sous-programme - entrée ; sorties ; retour, à partir d'un énoncé
- Utiliser de manière appropriée le vocabulaire lié aux langages Orientés Objet dans le cadre de java
- Exploiter quelques classes et packages existants
- Exploiter la Javadoc, la générer en utilisant les tags adaptés avec un contenu pertinent
- A partir de l'énoncé d'un problème, identifier, choisir et utiliser la structure de données appropriée parmi : Tableau, ArrayList
- Interpréter un diagramme de classes fourni, nommer les éléments constituant le diagramme de classes, produire le code java correspondant
- Pratiquer l'anglais (lire et écrire dans un contexte technique)

AUTOMATIQUE ET TRAITEMENT DU SIGNAL

OBJECTIFS

- Être capable de modéliser un système linéaire monovisible, d'en analyser les performances afin de synthétiser un correcteur ou un filtre numérique répondant à un cahier des charges donné.

OBJECTIFS D'APPRENTISSAGE

- 1) Etablir la fonction de transfert d'un système mono-variable à temps continu / à temps discret.
- 2) A partir d'un énoncé, représenter une boucle de régulation / d'asservissement sous la forme d'un schéma-bloc.
- 3) Statuer sur la stabilité d'un système linéaire bouclé à partir du ou des critères appropriés (Nyquist, Routh, ...).
- 4) Quantifier les performances d'une boucle de régulation / d'asservissement en termes de stabilité, rapidité et précision.
- 5) Expliquer le rôle des actions élémentaires de correction P, I et D.
- 6) A partir d'un cahier des charges donné, régler les paramètres des correcteurs PID, à avance de phase et à retard de phase.
- 7) A partir d'un cahier des charges donné, calculer les paramètres puis établir la structure optimale de réalisation d'un filtre linéaire récursif ou non.
- 8) Analyser les réponses temporelle et fréquentielle d'un système linéaire bouclé / d'un filtre linéaire.

OUTILS SCIENTIFIQUES POUR L'INGENIEUR

❖ PHENOMÈNES ÉLECTROMAGNÉTIQUES

OBJECTIFS D'APPRENTISSAGE

Dimensionner, à partir d'un cahier des charges, un dispositif électromagnétique simple.

Utiliser les différents opérateurs mathématiques sur des problématiques électromagnétiques.

Identifier les différents régimes (magnétostatique, électrostatique et propagation)

Décrire les grandeurs constituant les équations de Maxwell

Expliquer les différents phénomènes présents dans un condensateur et résoudre ce problème électrostatique en intégrant les caractéristiques physiques des matériaux.

Expliquer les différents phénomènes présents dans un système magnétique et résoudre ce problème magnétostatique en intégrant les caractéristiques physiques des matériaux.

Expliquer le phénomène de propagation dans un guide d'onde et résoudre les équations de Maxwell pour obtenir les équations de propagation.

Résoudre les équations de propagation dans le cas d'une onde et réflexion de cette onde sur différents matériaux.

PROGRAMME

- Outils et opérateurs mathématiques.
- Introduction des différents régimes (magnétostatique, électrostatique et propagation)
- Introduction des équations de Maxwell et des lois constitutives.
- Etude d'un problème électrostatique : le condensateur
 - Equations de Maxwell en électrostatique
 - Notion de charges électriques
 - Le théorème Green-Ostrogradski
 - Le théorème de Gauss
 - Exemples d'applications
 - Relation entre l'excitation électrique et le champ électrique.
 - Energie électrostatique
 - Conditions de passage
 - Milieux diélectriques
- Etude d'un problème magnétique quasi-statique : le transformateur
 - Equations de Maxwell en magnétostatique
 - Le théorème de Stokes
 - Le théorème d'Ampère
 - Exemples d'applications
 - Relation entre l'excitation magnétique et le champ magnétique
 - Relation entre le flux magnétique et le champ magnétique.
 - Energie magnétique
 - Conditions de passage
 - Matériaux magnétiques
 - Modèle d'Hopkinson : Analogie avec un circuit électrique
- Etude d'un problème de propagation : Guide d'onde
 - Equations de Maxwell en régime de propagation
 - Equations de propagation des champs électrique et magnétique.
 - Résolutions des équations de propagation
 - Définir les paramètres associés à une onde
 - Vecteur et longueur d'onde
 - Polarisation
 - Vitesses de phase et de groupe
 - Introduction d'une onde plane progressive monochromatique (OPPM)
 - Réflexion d'une onde quelconque sur différents types de matériaux.

❖ MATHÉMATIQUES POUR L'INGÉNIEUR

OBJECTIFS D'APPRENTISSAGE

Acquérir des méthodes et des outils mathématiques nécessaires à un ingénieur pour la modélisation et la résolution des problèmes en génie électrique, électronique, automatique et génie des procédés.

PROGRAMME

- Transformée de Laplace
- Applications à la résolution d'équations différentielles
- Séries de Fourier
- Transformation intégrale de Fourier
- Applications à la résolution d'équations aux dérivées partielles
- Transformation en Z

❖ THÉORIE DES PROBABILITÉS

OBJECTIFS D'APPRENTISSAGE

Faire apparaître le caractère aléatoire des phénomènes physiques (signaux aléatoires...) Assurer au maximum leur mesure pour aider à la décision et évaluer le coût.

PROGRAMME

- Espaces probabilisés
- Variables aléatoires discrètes
- Principales lois discrètes
- Variables aléatoires continues
- Loi normale et théorème de la limite centrale
- Problèmes de convergence : Loi faible des grands nombres

❖ THERMIQUE

OBJECTIFS D'APPRENTISSAGE

Identifier les différents modes de transfert thermique (conduction, convection, rayonnement)

Connaitre les lois qui les régissent (Fourier, Newton, Planck et Stefan).

Ecrire le bilan thermique global ou local d'un système et ses conditions aux limites.

Utiliser la notion de résistance thermique (milieu passif, situation stationnaire)

Appliquer ces notions à des situations impliquant l'électricité.

PROGRAMME

- Notions fondamentales : température, flux, et densité de flux
- Lois fondamentales et conditions aux limites.
- Bilan thermique. Production d'énergie thermique
- Notion de résistance thermique. Application à l'habitat
- Analogies thermique/électrique.
- Ailettes : équation générale, dimensionnement, efficacité
- Dissipateurs pour composants électroniques.
- Régime instationnaire : nombre de Biot, nombre de Fourier
- Propagation thermique : diffusivité et effusivité
- Grandeurs caractéristiques du transfert radiatif
- Etalon de rayonnement : le corps noir
- Corps gris, corps gris par morceaux, émissivités spectrale et totale

- Absorptivités spectrale et totale. Effet de serre

OBJECTIF

Être capable de s'exprimer efficacement en français et de façon relativement correcte en anglais, à l'oral comme à l'écrit, dans des contextes professionnels, semi-professionnels ou familiaux.

❖ ANGLAIS

Être capable de s'exprimer en anglais de façon relativement correcte à l'oral comme à l'écrit dans des situations courantes et professionnelles familiales.

OBJECTIFS D'APPRENTISSAGE

A l'issue du module, les étudiants seront capables de :

- Maîtriser un vocabulaire professionnel de base (l'entreprise, l'environnement de travail, la bureautique)
- Demander et échanger des informations sur des sujets familiaux en démontrant une assez bonne maîtrise grammaticale et syntaxique (formes affirmatives, interrogatives, l'utilisation des auxiliaires)
- Prendre part à une conversation sur des sujets familiaux
- S'exprimer sur ses expériences en utilisant de façon assez systématique les temps verbaux appropriés
- Expliquer de façon simple la structure d'une entreprise et décrire ses activités
- Rédiger des documents professionnels simples (CV, courriel) en se basant sur un modèle et en respectant les conventions du pays cible
- Comprendre les points essentiels d'un document écrit ou oral quand un langage clair et standard est utilisé et qu'il s'agit de sujets familiaux

❖ COMMUNICATION

Être capable de s'exprimer efficacement à l'oral comme à l'écrit dans des contextes professionnels et semi-professionnels.

OBJECTIFS D'APPRENTISSAGE

A l'issue du module, l'étudiant sera capable de :

- Transmettre clairement un message, notamment en utilisant la bonne terminologie, le bon registre, des articulateurs, des connecteurs, tout en faisant preuve d'une bonne correction syntaxique et grammaticale.
- Rédiger des écrits professionnels comme des mails et des lettres formelles, des ordres du jour, des comptes rendus de réunions et des rapports.
- Prendre la parole en public, à l'aide d'un support visuel si le cadre l'exige, pour présenter le fruit de ses recherches ou de son travail dans le cadre d'un exposé ou d'une soutenance.

- Rechercher de façon pertinente des informations, des sources documentaires, de les exploiter de façon rigoureuse et dans le respect de la propriété intellectuelle, et de les référencer dans le corps du texte et dans une bibliographie.

SEMESTRE 6

COMMUNICATION LIFI

OBJECTIFS

Concevoir et réaliser une transmission sans fil utilisant la lumière visible

OBJECTIFS D'APPRENTISSAGE

- Extraire des informations utiles à partir d'une documentation
- Analyser un montage, calculer les éléments
- Concevoir un montage à partir d'un cahier des charges, choisir les composants
- Câbler puis valider un montage en effectuant des mesures pertinentes

UE(S) PRE REQUIS

- Théorèmes de base
- Pratique de la mesure
- Montages de base de l'ampli op idéal
- Filtrage
- Utilisation de pspice temporel et fréquentiel

PROGRAMMER AVEC JAVA – S6

OBJECTIF

A partir d'un énoncé pouvant inclure un diagramme de classes UML (5 classes environ liées par des relations d'association et d'héritage), écrire, tester et documenter une application Java incluant :

- une base de données
- une interface graphique
- des algorithmes incluant au maximum 4 structures imbriquées.

OBJECTIFS D'APPRENTISSAGE

- A partir d'un énoncé, choisir les structures de contrôles, déterminer les variables, et la séquence d'instruction permettant de produire le résultat attendu (à l'écrit)
- Utiliser de manière appropriée le vocabulaire :
 - Des langages Orientés Objet (dans le cadre de Java)
 - Des Systèmes de Gestion de Bases de Données Relationnelles
- Exploiter des classes et des packages existants
- Exploiter la Javadoc, la générer en utilisant les tags adaptés avec un contenu pertinent
- A partir de l'énoncé d'un problème, identifier, choisir et utiliser la structure de données appropriée parmi : Tableau, ArrayList, Fichiers texte

- Interpréter un diagramme de classes fourni, nommer les éléments constituant le diagramme de classes, produire le code java correspondant
- Créer une fenêtre graphique avec quelques composants et interactions avec l'utilisateur
- Expliquer à quoi sert le mécanisme de capture d'une exception
- Formuler une requête sur une table, une jointure (SELECT) ainsi que les instructions de manipulation de données (INSERT, UPDATE, DELETE)
- Expliquer le rôle de JDBC
- Adapter un exemple fourni pour exécuter des instructions SQL dans une application java
- Pratiquer l'anglais (lire et écrire dans un contexte technique)

CONVERSION DE L'ÉNERGIE ÉLECTRIQUE

OBJECTIFS

Dans un contexte de conversion de l'énergie, les étudiants seront amenés à maîtriser le fonctionnement d'un moteur, agir sur sa vitesse, savoir l'alimenter à travers des convertisseurs à partir d'un réseau électrique afin de les préparer à concevoir une chaîne de conversion d'énergie.

OBJECTIFS D'APPRENTISSAGE

- Être capable de proposer une méthode de mesure de puissance d'une installation triphasée (ou monophasée) et analyser les résultats.
- Être capable de choisir un convertisseur en fonction d'une application industrielle nécessitant du continu.
- Être capable de choisir le type de composants intervenants dans un convertisseur de puissance.
- Être capable de d'expliquer et justifier le principe de fonctionnement d'un moteur électrique (MCC et MAS) et d'un transformateur monophasé.
- Être capable d'agir sur la vitesse ou le couple à partir des paramètres du modèle dans le cas de la MCC.
- Être capable d'identifier les paramètres du modèle du transformateur monophasé afin de calculer son rendement.
- Être capable d'expliquer le principe de fonctionnement d'un transformateur et de dresser son bilan de puissance.

OBJECTIFS D'APPRENTISSAGE

L'objectif du module est de sensibiliser les étudiants aux notions de systèmes numériques en reprenant de façon simple et accessible les principes de la logique combinatoire et séquentielle, ainsi que d'avoir une première approche du développement sur un système électronique piloté par microcontrôleur.

PROGRAMME

- Analyser un circuit électronique intégrant des fonctions logiques
 - niveau 0: Décrire et citer les fonctions logiques de base
 - niveau 1: Décrire le comportement d'un circuit simple à base de logique combinatoire
 - niveau 2: Même chose avec des éléments séquentiels
- Synthèse de circuits logiques
 - niveau 0: Décrire les outils et méthodes liées à la synthèse de fonctions logiques
 - niveau 1: Appliquer les bons outils à la synthèse d'un circuit logique combinatoire (4 variables maximum).
 - niveau 2: Appliquer les bons outils à la synthèse d'un circuit logique séquentiel (2 bascules D ou JK).
 - niveau 3: Concevoir un circuit logique à partir d'un cahier des charges.
 - niveau 4: Corriger un circuit avec erreur
 - niveau 5: Critiquer et améliorer un circuit
- Machines à états (Réseau de Pétri)
 - niveau 0: Citer et décrire les formalismes associés aux machines à états
 - niveau 1: Décrire le fonctionnement d'un système à état simple (x états/transitions) formalisé.
 - niveau 2: Concevoir une machine à états formalisée à partir d'un cahier des charges simple
- Utilisations et traitements des nombres en bases binaires
 - niveau 0: Effectuer les conversions entre les bases principales utilisées en numérique. (2, 16, 10)
 - niveau 1: Appliquer les règles de calcul liées aux représentations entières signées ou non signées.
 - niveau 2: Appliquer les règles de calculs logiques basées sur les opérateurs classiques.
- Décrire et mettre en œuvre un système embarqué
 - niveau 0: Décrire les éléments essentiels d'une architecture embarquée (capteurs, microcontrôleur, programme, actionneurs)
 - niveau 1: Analyser des caractéristiques techniques et en déduire des contraintes d'utilisation.
 - niveau 2: Mettre en œuvre un système de développement autour d'une carte électronique afin de réaliser une application embarquée (simulation)

- niveau 3: Mettre en œuvre un système de développement autour d'une carte électronique afin de réaliser une application embarquée (carte réelle)
- Logique côté pratique (circuits, niveaux, outils, simulation, câblage, ...)
 - niveau 0: Reproduire un schéma électronique logique simple sous forme informatique et réelle et vérifier le comportement du circuit.
 - niveau 1: Utiliser l'outil informatique pour simuler le comportement d'un circuit conçu pour répondre à un cahier des charges simple.
 - niveau 2: Utiliser un analyseur logique pour détecter et corriger un défaut sur un circuit logique fourni.

❖ **ANGLAIS**

Être capable de communiquer en anglais à l'oral comme à l'écrit avec une assurance grandissante sur des sujets familiers ou préparés à l'avance.

OBJECTIFS D'APPRENTISAGE

A l'issue du module, les étudiants seront capables de :

- Maîtriser le vocabulaire lié aux voyages d'affaires et aux présentations professionnelles
- Faire une présentation en groupe, préparée à l'avance, d'un sujet non technique qui peut être suivi par l'auditoire sans difficulté la plupart du temps, et dans lequel les points importants sont suffisamment explicités
- Comprendre et répondre avec une certaine aisance aux questions posées à l'issue de la présentation
- Rédiger un document court avec un niveau de correction grammaticale et lexicale suffisant pour que le sens général reste clair
- Produire un document écrit en respectant la propriété intellectuelle et en citant les sources bibliographiques selon les consignes données
- Comprendre un document écrit ou oral quand un langage clair et standard est utilisé et qu'il s'agit de sujets familiers
- Renforcer et étendre les bases grammaticales

❖ GESTION FINANCIÈRE

OBJECTIFS D'APPRENTISSAGE

Acquérir les fondements de la comptabilité générale et de l'analyse financière.

Comprendre comment s'élaborent les états financiers d'une entreprise, le bilan et le compte de résultats. Être capable de les lire, de les interpréter et d'avoir un regard critique.

Être capable de poser un diagnostic financier pour identifier si l'entreprise est viable et rentable,

Être capable de comprendre comment l'entreprise se finance.

Être capable de réaliser une analyse statique (fonctionnelle et financière) et une analyse dynamique (à partir d'un ou plusieurs bilans).

PROGRAMME

- la nécessité d'une information financière pour l'ensemble des parties prenantes avec des besoins distincts
- Le fonctionnement des comptes et le plan de comptes
- L'organisation comptable
- Les techniques de l'enregistrement des opérations comptables.
- L'élaboration du bilan et du compte de résultat
- Les étapes du diagnostic
- Analyse de l'activité et de la rentabilité : soldes intermédiaires de gestion (SIG), capacité d'autofinancement (CAF par les deux méthodes de calcul), la rentabilité économique et la rentabilité financière (l'effet de levier/l'effet de massue).
- Analyse de l'équilibre financier : le bilan fonctionnel (construction, calcul du FR, BFR, BFRE et BFRHE), les flux de trésorerie (tableau de financement, ex du tableau OEC) et les principaux ratios de structure financière ainsi que leur signification.

SEMESTRE 6

MODULES ÉLECTIFS DE CULTURE GÉNÉRALE

❖ DÉVELOPPEMENT DURABLE : NUMÉRIQUE RESPONSABLE

OBJECTIFS D'APPRENTISSAGE

- Comprendre l'impact environnemental du numérique.
- Apprendre à mettre en œuvre des pratiques de numérique responsable.
- Sensibiliser à l'éthique et à l'inclusion dans l'utilisation du numérique.

PROGRAMME

1. **L'état de la planète** : Pourquoi le numérique responsable est crucial.
2. **Les impacts du numérique** : Exploration de l'empreinte environnementale du numérique.
3. **Nécessité de Changer** : Pourquoi et comment modifier nos pratiques numériques.
4. **Le Numérique Responsable** : Définitions et mise en œuvre de pratiques responsables.
5. **Agir dès maintenant - Introduction** : Premiers pas vers le changement.
6. **Le poste de travail** : Optimisation de l'utilisation des équipements numériques.
7. **Le système d'impression** : Pratiques responsables en matière d'impression.
8. **L'architecture Web** : Construction de sites web durables.
9. **Le Data center et le cloud** : Amélioration de l'efficacité des data centers et services cloud.
10. **Les services numériques et bureautiques** : Utilisation responsable des outils numériques.
11. **Conclusion générale** : Résumé des apprentissages et prochaines étapes.
12. **Glossaire** : Termes clés et définitions.
13. **Certificat de connaissance** : Évaluation des connaissances acquises.

MASSES HORAIRES, COEFFICIENTS ET CONTROLES
 2ème année - Semestres 7 et 8
 FILIERE INITIALE STATUT APPRENTI
 ANNEE 2024/2025

UE Semestre 7	Réf.	Module	SEM.	COURS	TD	TP	TUT	APP	EXAMEN	TOTAL	COEF	ECTS	CONTROLES
ELECTRONIQUE	EL2C3-F	Modulations	7	1		16			1	18	5	5	CCTP 50%, CE 50%
	EL2C4-F	Numérisation	7	1		16			1	18	5	2	CCTP 40%, CE 40% QCM 20%
INFORMATIQUE	IN2C1-F	Cybersécurité	7		4	12				16	3	2	CC
	IN2C2-F	Réseaux informatiques et systèmes d'exploitation	7		10	20			2	32	7		1 CE (50%) , CCTP (25%), CITP(25%)
SYSTEMES EMBARQUES	SE2C1-F	Systèmes à microprocesseurs	7	5		15	26	18	3	67	7	3	2CE/1 CCTP/ 1 CTP (modalités particulières)
	LG2C1-A	Anglais (50%)	7		24					24	10	2	CC1 oral (35%), CC2 écrit 1 (35%), CC3 écrit 2 (20%), CC4 Production/Participation (10%)
LANGUES ET GESTION	LG2C3-F	Projet Voltaire (20%)											CC
	LG2C2-F	Management de projet Agile (30%)	7	6	6				1	13			CE
		Transition Écologique pour un Développement Soutenable	7	18					2	20	3	2	CE
TEDS		TRAVAIL EN AUTONOMIE	7							20			
										228	40	11	

PARCOURS TECHNOLOGIQUES													
UE Semestre 7	Réf.	Module	SEM	COURS	TD	TP	APP	EXAMEN	TOTAL	COEF	ECTS	CONTROLES	
Parcours INFORMATIQUE	PIN201-F	Conception de sites web	7	4	2	22		2	30	10	3	1 CE (70%) / 1 CCTP(30%)	
	PIN202-F	Introduction au DevOps	7	10	2	16		2	30	10	3	1 CE (70%) / 1 CCTP (30%)	
									60	20	3		
Total semestre 7										288	60	14	

Total semestre 7

UE Semestre 8		Module	SEM.	COURS	TD	TP	TUT	APP	EXAMEN	TOTAL	COEF	ECTS	CONTROLES
ENSEIGNEMENTS TRANSVERSAUX	ET2C1-A	Anglais (60%)	8		18					18	10	2	CC1 oral (35%), CC2 écrit 1 (35%), CC3 écrit 2 (20%), CC4 Production/Participation (10%)
	ET2C2-F	Electifs "Culture d'entreprise"(40%)	8	18					2	20			CE
	ETC3-F	Atelier RH	8		3					3			
Dominantes		Dominantes	8							180	30	12	
										221	40	14	

ENTREPRISE		Entreprise	7 & 8									27	Rapport/Soutenance + Evaluation pratique par l'Entreprise
		Mission à l'étranger	8									5	Rapport/Soutenance
												32	
CC : Contrôle Continu / CE : Contrôle Ecrit / CO : Contrôle Oral / CCTP : Contrôle Continu de TP / CTP : Contrôle Individuel de TP										Total semestre 8			
										221	40	46	
TOTAL										509	100	60	

(*) Un électif est à choisir parmi les suivants : Stratégie d'Entreprise

SEMESTRE 7

ÉLECTRONIQUE

❖ MODULATIONS ET NUMÉRISATION

OBJECTIFS D'APPRENTISSAGE

Expliquer le principe et l'intérêt des modulations d'amplitude et de fréquence, reconnaître leurs chronogrammes et spectres associés

Reproduire l'architecture d'un récepteur, étape indispensable pour aborder les modulations numériques, choisir les composants adéquats

Câbler une chaîne complète modulateur + récepteur/démodulateur et interpréter les résultats.

Maîtriser les principes de conversions AN et NA.

Se familiariser avec les principales architectures des CAN et CNA disponibles sur le marché.

Savoir faire un choix de CAN et CNA en fonction d'une application donnée.

Savoir tester un CAN et un CNA pour vérifier son bon fonctionnement.

CONTENU

Modulations :

- Les modulations et démodulations d'amplitudes
- La QAM analogique
- Le récepteur à changement de fréquence
- La modulation de fréquence, le VCO

Numérisation :

- Échantillonnage, limite de Shannon, quantification
- Le CAN, le CNA, mise en œuvre
- La chaîne complète CAN + modulation + démodulation + CNA

INFORMATIQUE

❖ CYBERSÉCURITÉ

OBJECTIFS D'APPRENTISSAGE

Décrire les enjeux de la cybersécurité, expliquer le besoin de sécurité pour les systèmes d'information, définir les principales menaces et leur typologie

Appréhender et adopter les règles d'hygiène de base de la cybersécurité

Lister et décrire quelques solutions techniques de sécurité

PROGRAMME

1. Notions générales
 - a. Données personnelles / données d'entreprise
 - b. Système d'information
 - c. Propriété de sécurité
 - d. Notions de menaces, vulnérabilité, attaques
2. Attaques et failles
3. Panorama de quelques menaces
4. Utilisateurs et contrôle d'accès
5. Cryptographie
6. Le masquage des données
7. Sécurisation d'un réseau
8. Intégrer la sécurité au sein d'une organisation

❖ RÉSEAUX INFORMATIQUES ET SYSTÈMES D'EXPLOITATION

OBJECTIFS

Concevoir un réseau répondant à un ensemble de critères imposés (nombre de segments logiques, nombre de machines par segment logique, nature du nœud d'interconnexion à mettre en place...), en définir le plan d'adressage IP complet et valider son fonctionnement au moyen d'un simulateur réseau ; mettre en pratique les connaissances réseau pour paramétrer la configuration réseau des machines Linux et Windows ; manipuler les commandes de base du système Linux.

OBJECTIFS D'APPRENTISSAGE

- Choisir le ou les équipements (concentrateur, commutateur, routeur) à employer pour répondre à un cahier des charges donné (réseau local, interconnexion de réseaux...),
- Démontrer l'intérêt d'un modèle en couches (OSI, TCP/IP) dans un contexte de systèmes ouverts, expliquer le rôle d'un protocole dans le cadre d'une communication réseau, expliquer de manière précise les concepts d'adressage physique (MAC) et logique IPv4, expliquer le rôle des différents champs des entêtes Ethernet (niveau 2 OSI), IP (niveau 3 OSI) et transport (niveau 4 OSI) et interpréter la valeur de chacun de ces champs (analyse d'une trame, reconstitution d'un échange...),
- Analyser les trames échangées des protocoles IP, ICMP, ARP,
- Simuler un réseau informatique,
- Décrire l'utilité des principaux protocoles applicatifs utilisés sur internet (HTTP, FTP, DNS, SMTP-POP3/IMAP, Telnet) et expliquer la nécessité d'un port dédié à chaque application sur chaque machine,
- De configurer les paramètres réseau d'une machine opérant sous Linux et sous Windows,
- Ecrire des scripts simples pour Linux pour automatiser certaines tâches,
- Expliquer la relation entre le système d'exploitation et le réseau.

PROGRAMME

- Principes de téléinformatique (structures des réseaux, supports de communication, caractéristiques de la transmission)
- Les modèles OSI et TCP/IP (intérêt, concepts et protocoles)
- Les composants matériels des réseaux (couche physique)
- Les technologies LAN (couche liaison de données)
- Principes d'adressage IPv4
- Les protocoles de base de la couche réseau (ARP, RARP, BOOTP, IP, ICMP)
- Les commandes de base Linux

SYSTÈMES EMBARQUÉS

❖ SYSTÈMES À MICROPROCESSEURS

OBJECTIFS D'APPRENTISSAGE

- Être capable d'extraire les informations pertinentes d'une documentation technique en anglais.
- Citer des exemples d'applications intégrant de l'électronique programmable embarquée.
- Décrire l'architecture interne d'un microprocesseur ainsi que le rôle des grandes fonctions qui le constituent.
- Etablir les relations entre un mapping mémoire et l'implémentation d'une application logicielle dans un système.
- Décrire le rôle d'un petit programme simple (moins de 20 lignes) écrit en assembleur et/ou en langage C et analyser les éventuels problèmes de calculs ou de fonctionnement qu'il pose.
- Expliquer les différences entre les types de données (entier, flottant, signé, non signé, ASCII) gérés dans les applications à microprocesseurs.
- Concevoir, écrire et tester un petit programme en langage C destiné à être exécuter sur une carte microprocesseur
- Utiliser des bibliothèques de fonctions fournies pour réaliser une application en suivant un cahier des charges donné.
- Imaginer, concevoir et réaliser un démonstrateur en fonction de l'expression d'un besoin.
- Présenter à un public de manière synthétique le résultat d'un travail technique.
- Mettre en œuvre, à partir de la documentation d'un μC , les périphériques suivants : Timer, convertisseur analogique-numérique, entrées-sorties Tout Ou Rien.
- Concevoir, écrire et tester un programme en langage C répondant à un cahier des charges et faisant intervenir les périphériques cités ci-dessus.
- Être capable de transposer les notions abordées à d'autres μC et/ou périphériques.

PROGRAMME

- La place de l'électronique numérique dans le monde actuel
- La constitution d'un système embarqué
- Présentation de l'architecture générale des microprocesseurs
- Les signaux vitaux d'un microprocesseur

- Types et organisations des mémoires électroniques
- La gestion des échanges entre le processeur et les mémoires
- La gestion des données dans un microprocesseur
- Les niveaux de langage dans le développement d'une application
- Les éléments de bases de la programmation assembleur
- L'utilisation d'un environnement intégré de développement
- L'utilisation des fonctions de bases d'un débogueur
- L'écriture d'application en C sur un microcontrôleur
- Rôle des entrées-sorties
- Types d'interfaces
- Protocole d'utilisation en scrutation et en interruption
- Les entrées-sorties Tout Ou Tien
- La détection de fronts
- L'ADC
- Les Timers

LANGUES ET GESTION

❖ ANGLAIS

OBJECTIFS D'APPRENTISSAGE

Être capable de s'exprimer en anglais avec davantage d'aisance à l'oral comme à l'écrit dans des situations courantes et professionnelles familières.

PROGRAMME

1. Approfondir le vocabulaire professionnel de base (structure et activités d'une entreprise, l'environnement de travail) et renforcer les bases grammaticales
2. Être capable de s'exprimer à l'écrit sur ses expériences en entreprise de façon succincte et en démontrant une assez bonne maîtrise grammaticale et syntaxique
3. Être capable d'appréhender un entretien d'embauche en anglais : être capable de s'exprimer de façon simple sur son expérience professionnelle, sa formation, ses qualités et ses motivations
4. Être capable de rédiger une lettre de motivation en respectant les conventions du pays cible et avec un registre approprié
5. Être capable de comprendre les points essentiels d'un document écrit ou oral quand un langage clair et standard est utilisé et qu'il s'agit de sujets familiers

❖ MANAGEMENT MODERNE DE PROJETS

OBJECTIFS D'APPRENTISSAGE

Apprendre et appliquer les fondamentaux du management de projet

PROGRAMME

- Identifier les différentes approches du management de projet
- Maîtriser les bases du « langage » projet
- Identifier les typologies des projets
- Expliciter les objectifs d'un projet
- Définir le/les livrables d'un projet
- Identifier toutes les parties prenantes d'un projet
- Réaliser un budget prévisionnel
- Cadrer une réunion de projet, particulièrement la réunion de lancement du projet
- Approche traditionnelle
 - Réaliser un rétroplanning (diagramme de Gantt)
 - Identifier les risques et les freins d'un projet
 - Utiliser des outils d'aide à la décision (matrice, mind map, ...)
 - Connaître les compétences d'un chef de projet
 - Utiliser des outils modernes de gestion de projet et de travail collaboratif
- Approche agile
 - Comprendre les fondements et les spécificités de l'approche Agile (Manifeste Agile, 4 valeurs, 12 principes)
 - Les notions et ingrédients de base de l'Agilité (product owner, scrum master, product backlog, DoD, sprint, daily scrum, etc.)
 - Connaître le fonctionnement des principales méthodes agiles (Speed boat, scrum, kanban, rétrospective, travail en binôme, etc.)
 - Mettre en place une planification de projet basée sur les principes agiles
 - Gestion des risques, limites et dérives
 - Se repérer avec les outils de management visuel et d'avancement dans un contexte agile

SEMESTRE 7

PARCOURS TECHNOLOGIQUES

NB : Les apprentis ne choisissent qu'un parcours

PARCOURS INFORMATIQUE

❖ CONCEPTION DE SITES WEB

OBJECTIF

Créer et déployer un site web responsif dynamique et accessible, respectant les principes de séparation du contenu et de la présentation, valide selon les normes HTML5 et CSS du w3c et sécurisé contre les injections SQL et attaques par défaçage.

ACQUIS D'APPRENTISSAGE VISES

Expliquer :

- La relation entre Internet et le Web
- Ce qu'est HTTP
- La notion de client et de serveur web
- La notion de page adaptative
- Le rôle des langages serveur (PHP, Python, JEE, ...)
- Le rôle des langages client (HTML, CSS, Javascript, ...)
- La différence entre un rendu côté serveur et côté client
- Les étapes majeures de la réalisation d'un projet web
- L'importance de la validation pour la sécurité d'un site web

Créer et déployer un site web :

- Dynamique, adaptatif et personnalisé,
- Respectant les principes de séparation du contenu et de la présentation,
- Respectant les bonnes pratiques pour l'accessibilité
- Valide selon les normes HTML5 et CSS du w3c,
- Sécurisé contre les injections SQL et attaques par défaçage,
- En mode projet, exploitant notamment l'outil de gestion de versions décentralisé Git et des outils de déploiement/intégration continue.

PROGRAMME

Le cours développe les notions suivantes :

- Design adaptatif d'une page web
- Relation entre Internet et Web, HTTP, clients/serveur, les langages du Web

- Le concept d'API
- Traitement de formulaires et pages dynamiques avec un langage serveur
- Pages personnalisables avec sessions côté serveur et bases de données

Les travaux pratiques traitent :

- HTML : création de liens hypertextes, insertion d'images
- CSS/JavaScript : mise en forme et animation d'un document
- Codage d'une page web responsive avec un framework front-end Javascript
- Validation du respect des contraintes d'accessibilités à l'aide d'outils dédiés

Le projet est composé de 3 phases :

- Réalisation d'un dossier de conception
- Création et déploiement du site
- Présentation du site hébergé sur un serveur

❖ DEVOPS / CYCLES DE DÉVELOPPEMENT AGILE

OBJECTIFS D'APPRENTISSAGE :

- Connaître les étapes de la chaîne de déploiement automatisé d'une application
- Savoir mettre en place une chaîne de déploiement et de test via un outil de CI/CD
- Savoir architecturer une application en vue d'un déploiement local et Cloud
- L'utilisation de Machines Virtuelles / Conteneurs dans le cadre du CI/CD
- Déployer une application sur une infrastructure Cloud (IaaS, IaaS)
- Comprendre l'intégration des outils et méthodes dans un workflow Agile

PROGRAMME

- Introduction
 - Les méthodes historiques de déploiement et leurs contraintes
 - L'avènement de l'intégration continue
 - L'avènement du déploiement continu
 - Le concept de Pipeline
- Le versionning et son utilité pour le déploiement
- Les conteneurs (Docker)
 - Comprendre les concepts de base de Docker
 - Créer des images personnalisées
 - Utiliser les outils Docker Compose pour définir et gérer des applications multi-conteneurs.
 - Utiliser les outils d'intégration continue avec Docker pour automatiser les processus de déploiement
- Les tests
 - Le Test Driven Development

- La spécification OpenAPI et SwaggerUI
 - Les tests unitaires et leur automatisation
 - Les tests fonctionnels et leur automatisation
- Le déploiement
 - Local pour le cycle de développement
 - Distant pour le cycle de production
 - La sécurité en production
 - Sauvegarde, fiabilité et résilience

SEMESTRE 8

ENSEIGNEMENTS TRANVERSAUX

❖ ANGLAIS

OBJECTIFS D'APPRENTISSAGE

Être capable de s'exprimer en anglais avec davantage d'aisance à l'oral comme à l'écrit dans des situations courantes et professionnelles. Pour les cours thématiques, approfondir ses connaissances dans un domaine et s'exprimer en anglais sur un sujet défini.

ACQUIS D'APPRENTISSAGE VISES

1. Renforcer et approfondir les bases grammaticales
2. Etendre son vocabulaire (vie courante, monde du travail, finances)
3. Être capable de faire face à l'oral aux situations de tous les jours en utilisant des expressions idiomatiques fréquentes et un registre approprié
4. Être capable de comprendre une plus grande diversité de documents écrits
5. Être capable de comprendre des enregistrements avec des accents variés quand un langage clair et standard est utilisé et qu'il s'agit de sujets familiers
6. Se préparer à l'examen externe

❖ STRATÉGIE D'ENTREPRISE

OBJECTIFS D'APPRENTISSAGE

L'objectif du cours est d'apprendre aux étudiants les concepts de base de la stratégie d'entreprise. En partant du postulat que la stratégie d'une entreprise consiste en la détermination d'un objectif, un plan ou l'atteindre puis des actions à mener, le cours fera un focus particulier sur la concurrence, les stratégies de diversifications / croissance / désinvestissement, ainsi que la rentabilité des projets. Notamment et non limités aux outils suivants, les outils comme les 5 forces de Porter, le SWOT, cartes stratégiques et les VAN (Valeur Actuelle Nette) / TRI (Taux de Rentabilité Interne) seront définis et expliqués.

Finalement, Les problématiques liées à l'implémentation de ces stratégies sur le plan opérationnel sera abordé, d'un point de vue auditeur, en mettant en avant les erreurs communes observées en entreprise.

Le plan du cours pourra se présenter comme suit (sujet à changement):

- Introduction et définitions
- Deep dive sur les outils, leurs utilisations et limites (revues des éléments constituant les 5 forces de Porter et du SWOT)
- Les choix d'investissements
- Implémentation des stratégies, point de vue audit

Tout au long du cours, des cas d'école seront présentés pour illustrer le contenu du cours.

Les exclusions principales du cours seront les sujets marketing et ressources humaines.

❖ ATELIER RH

OBJECTIFS D'APPRENTISSAGE

Être capable :

- d'utiliser les outils de recherche de stages et d'emplois de façon optimisée
- de connaître les avantages et les dangers de l'e-réputation
- de construire un CV correct

PROGRAMME

Une conférence sur l'optimisation des outils de recherche et sur l'e-réputation.
Un atelier CV animé par des RH d'entreprises.

SEMESTRE 8 – DOMINANTE DÉVELOPPEMENT LOGICIEL TEST & QUALITÉ

ANALYSE ET CONCEPTION

❖ ALGORITHMIQUE ET STRUCTURES DE DONNÉES

OBJECTIFS D'APPRENTISSAGE

- **Comprendre les principes fondamentaux de l'algorithmique** : Apprendre à concevoir, analyser, et optimiser des algorithmes pour résoudre divers problèmes informatiques.
- **Maîtriser les structures de données classiques** : Acquérir une bonne connaissance des différentes structures de données (listes, piles, files, arbres, graphes, etc.) et comprendre leurs utilisations spécifiques.
- **Développer des compétences en programmation** : Améliorer la capacité à implémenter des algorithmes et des structures de données en utilisant un langage de programmation approprié.
- **Analyser la complexité des algorithmes** : Apprendre à évaluer l'efficacité des algorithmes, notamment en termes de temps d'exécution et de consommation de mémoire.
- **Résoudre des problèmes pratiques** : Appliquer des connaissances théoriques à des cas pratiques pour renforcer la compréhension et la capacité à résoudre des problèmes réels.

PROGRAMME

1. **Introduction à l'algorithmique**
 - Définitions et importance
 - Exemples d'algorithmes simples
2. **Complexité des algorithmes**
 - Notions de base de la complexité temporelle et spatiale
 - Analyse de la complexité des boucles, récursions, et opérations courantes
3. **Structures de données linéaires**
 - Listes chaînées
 - Piles et files
 - Tableaux et matrices
4. **Structures de données non linéaires**
 - Arbres (binaires, AVL, Rouge-Noir)
 - Graphes (représentations, parcours en profondeur et en largeur)
 - Tables de hachage
5. **Algorithmes de tri et de recherche**
 - Tri à bulles, tri par insertion, tri rapide, tri par fusion
 - Recherche linéaire, recherche binaire
6. **Algorithmes sur les graphes**
 - Algorithmes de Dijkstra, Bellman-Ford
 - Algorithme de Kruskal, de Prim
7. **Applications pratiques et études de cas**
 - Études de cas sur des problèmes de tri, de recherche de chemin, et d'optimisation

- Projet pratique: implémentation d'une application utilisant diverses structures de données

❖ GÉNIE LOGICIEL : UML ET DESIGN PATTERN

OBJECTIFS D'APPRENTISSAGE :

- **Comprendre les fondamentaux de l'UML (Unified Modeling Language)** : Apprendre à utiliser l'UML pour modéliser des systèmes informatiques, en incluant les différents types de diagrammes (cas d'utilisation, séquence, classes, etc.).
- **Maîtriser les diagrammes UML les plus courants** : Être capable de créer et d'interpréter des diagrammes de classes, de séquences, et de cas d'usage pour la conception de logiciels.
- **Découvrir les Design Patterns** : Connaître et comprendre les différents modèles de conception (patterns) pour résoudre des problèmes récurrents en génie logiciel.
- **Appliquer les Design Patterns dans des projets de développement** : Savoir choisir et implémenter le pattern adéquat en fonction des besoins spécifiques d'un projet.
- **Analyser et critiquer des architectures logicielles** : Évaluer l'efficacité et la pertinence des choix de modélisation et de patterns dans des études de cas.

PROGRAMME :

- **Introduction à l'UML**
 - Les différents types de diagrammes UML et leurs utilisations.
 - Pratique de modélisation avec des outils de création de diagrammes UML.
- **Diagrammes de Cas d'Utilisation**
 - Comprendre et identifier les acteurs d'un système.
 - Modéliser les interactions principales via les cas d'utilisation.
 - Ateliers pratiques sur des études de cas réels.
- **Diagrammes de Classes**
 - Structure et éléments d'un diagramme de classes : Exploration des composants clés tels que les classes, les relations, les interfaces, et les héritages.
- **Diagrammes de Séquences**
 - Modélisation des interactions entre objets : Utilisation des diagrammes de séquences pour illustrer le flux d'exécution et la communication entre objets dans un scénario donné.
- **Introduction aux Design Patterns**
 - Présentation des catégories de patterns : création, structuration, comportement.
 - Étude détaillée de patterns sélectionnés comme Singleton, Factory, Observer, et Strategy.
 - Application de ces patterns dans des mini-projets.
- **Études de Cas et Projet Final**
 - Analyse de systèmes existants et critique de leur conception.
 - Projet : conception d'une application en utilisant UML et plusieurs design patterns.

❖ BASES DE DONNÉES : SQL ET NOSQL

OBJECTIFS D'APPRENTISSAGE :

- Comprendre les différences fondamentales et les cas d'usage de SQL et NoSQL pour le développement de bases de données.
- Savoir choisir entre une base de données SQL ou NoSQL en fonction des besoins spécifiques d'un projet.
- Développer des compétences avancées en conception et interrogation de bases de données SQL et NoSQL.
- Appliquer les concepts appris en première année sur Oracle SQL à d'autres systèmes de gestion de bases de données.

PROGRAMME :

- **Introduction aux bases de données**
 - Révision des concepts de base de données SQL : modèles, schémas, normalisation.
 - Introduction aux bases de données NoSQL : types, avantages, quand les utiliser.
- **Bases de données SQL avancées**
 - Conception avancée de bases de données SQL : techniques de partitionnement, indexation et optimisation des requêtes.
 - Utilisation de SQL pour l'analyse de données : fonctions d'agrégation, requêtes complexes.
- **Bases de données NoSQL**
 - Types de bases de données NoSQL : document, clé-valeur, colonne, graphique.
 - Modélisation de données pour NoSQL : défis spécifiques et meilleures pratiques.
 - Cas pratiques d'utilisation de bases de données NoSQL : MongoDB, Redis, Cassandra.
- **Ateliers pratiques et études de cas**
 - Migration d'une application de SQL à NoSQL : étude de faisabilité, planification et exécution.
 - Conception et développement d'une application utilisant MongoDB pour la gestion de documents.
 - Analyse de performance et optimisation des requêtes dans un environnement SQL et NoSQL.

❖ DÉVELOPPEMENT LOGICIEL ET DEVOPS

OBJECTIFS D'APPRENTISSAGE :

- Approfondir la compréhension des méthodes et outils de CI/CD dans un contexte de déploiements fréquents et automatisés.
- Maîtriser l'orchestration de conteneurs pour optimiser les environnements de production.
- Développer des compétences avancées en matière de surveillance et de logging dans les environnements de production.
- Intégrer et optimiser les pratiques de sécurité dans le pipeline de déploiement.
- Comprendre et appliquer les principes de l'infrastructure en tant que code pour la gestion dynamique des ressources.

PROGRAMME :

- **Révision et Approfondissement des Fondamentaux**
 - Révision sur CI/CD : Importance et mise en œuvre pratique.
 - Discussion approfondie sur Docker et Kubernetes : de la configuration à l'orchestration.
 - Pratiques avancées en matière de versionnage avec Git et gestion de configurations.
- **Sécurité et DevOps**
 - Intégration des pratiques de sécurité : DevSecOps.
 - Automatisation des tests de sécurité et intégration dans le pipeline CI/CD.
 - Gestion des secrets et des accès dans un environnement DevOps.
- **Surveillance et Logging**
 - Outils et techniques de surveillance : Prometheus, Grafana.
 - Logging centralisé avec ELK Stack ou Fluentd : configuration et gestion.
 - Alerting et monitoring : Configuration pour une réactivité maximale.
- **Infrastructure as Code (IaC)**
 - Principes et outils principaux : Terraform, Ansible.
 - Scénarios pratiques d'utilisation d'IaC dans le cloud et on-premise.
 - Gestion des configurations et automatisation des déploiements avec IaC.
- **Projets Pratiques et Études de Cas**
 - Mise en place d'un pipeline CI/CD complet pour une application microservices.
 - Déploiement et gestion d'une application en utilisant Kubernetes et IaC.
 - Études de cas sur la résolution de problèmes dans des environnements de production complexes.

❖ PROGRAMMATION ORIENTÉE OBJET

OBJECTIFS D'APPRENTISSAGE :

- **Comprendre les principes de la programmation orientée objet en C++** : Adapter et appliquer les concepts appris en Java (classes, héritage, polymorphisme, etc.) au langage C++.
- **Maîtriser les spécificités de C++** : Apprendre à utiliser les particularités de C++ telles que les pointeurs, la gestion de la mémoire dynamique, les références, et les templates.
- **Implémenter des algorithmes et des structures de données en C++** : Appliquer les connaissances en algorithmique et structures de données pour implémenter des solutions optimisées en C++.
- **Développer des compétences en optimisation et analyse de performance** : Analyser et optimiser le code C++ en termes de temps d'exécution et de consommation de mémoire.
- **Appliquer les concepts à des projets pratiques** : Développer des applications concrètes en utilisant la programmation orientée objet en C++.

PROGRAMME :

- **Bases de la programmation en C++**
 - Variables, types de données, et opérateurs
 - Instructions de contrôle : conditions, boucles
 - Fonctions et portée des variables
- **Programmation Orientée Objet en C++**
 - Classes et objets
 - Constructeurs et destructeurs
 - Héritage et polymorphisme
 - Encapsulation et abstraction
- **Spécificités de C++**
 - Pointeurs et références
 - Gestion de la mémoire dynamique (new, delete)
 - Surcharge d'opérateurs
 - Templates et programmation générique
- **Structures de données en C++**
 - Implémentation des listes chaînées
 - Piles et files
 - Tableaux dynamiques et vecteurs
 - Arbres binaires et arbres équilibrés (AVL, Rouge-Noir)
 - Graphes : représentations et algorithmes de parcours
- **Algorithmes en C++**
 - Algorithmes de tri : tri à bulles, tri par insertion, tri rapide, tri par fusion
 - Algorithmes de recherche : recherche linéaire, recherche binaire
 - Algorithmes sur les graphes : (e.g : Dijkstra, Bellman-Ford, Kruskal, Prim)

❖ TDD ET QUALITÉ DU CODE

OBJECTIFS D'APPRENTISSAGE :

- **Maîtriser le développement basé sur les tests (TDD)** : Apprendre à développer des logiciels en écrivant les tests en amont de la phase de programmation, afin de garantir leur bon fonctionnement.
- **Améliorer la qualité du code** : Écrire un code propre, lisible et maintenable, qui respecte les normes et les bonnes pratiques de développement.
- **Automatiser les tests et la validation** : Utiliser des outils d'automatisation des tests pour assurer la qualité continue du logiciel tout au long du cycle de développement.
- **Intégrer les principes du Green IT dans le développement** : Sensibiliser sur les impacts environnementaux du développement logiciel et promouvoir des pratiques plus durables.

PROGRAMME :

- **Introduction au TDD**
 - Définition et principes fondamentaux
 - Avantages et défis de l'approche TDD
- **Cycle de vie du développement logiciel avec TDD**
 - Étapes du développement basé sur les tests
 - Intégration de TDD avec les méthodes agiles, comme SCRUM
- **Écriture de tests efficaces**
 - Techniques pour écrire des tests clairs et compréhensibles
 - Utilisation de frameworks de tests comme JUnit pour Java
- **Refactoring et amélioration du code**
 - Techniques de refactoring pour améliorer la lisibilité et la maintenabilité du code
 - Identification des "code smells" et des méthodes pour les éliminer
- **Outils de qualité du code**
 - Présentation d'outils comme SonarQube pour surveiller la qualité du code
 - Configuration et interprétation des rapports générés par ces outils
- **Pratiques de développement durable et Green IT**
 - Évaluation de l'impact environnemental du développement logiciel
 - Stratégies pour minimiser cet impact dans les projets de développement

MASSES HORAIRES, COEFFICIENTS ET CONTROLES
2ème année Dominante Intelligence Artificielle et Big Data (IA&BD)
Année 2024/2025

UE	Réf.	Module	SEMESTRE	COURS	TD	TP	EXAMENS	TOTAL	COEF. MOD	ECTS	CONTROLES
Mathématiques et Intelligence Artificielle	IABD-201-A	Mathématiques et statistiques appliquées à l'IA	8	12	0	12	1	25	4	5	1 CC/1 CE
	IABD-202-A	Modélisation mathématique pour l'IA	8	12	0	12	1	25	4		1 CC/1 CE
	IABD203-A	Machine Learning et Deep Learning	8	12	0	12	1	25	4		1 CC/1 CE
	IABD204-F	Architecture Big Data et Cloud Computing	8	12	0	12	1	25	4	5	1 CC/1 CE
Big Data & Cloud Computing	IABD205-F	Base de données Big Data avec NoSQL	8	12	0	12	1	25	4		1 CC/1 CE
	IABD206-A	Cloud Infrastructure Fundamentals Google Cloud : Associate Cloud Engineer	8	8	4	12	1	25	4		1 CC (conversion du score de validation selon les critères d'obtention de la certification) / 1 CE
PROJET IA	IABD207-F	Projet IA	8	4	0	26		30	6	2	1 CC
				72	4	98	6	180	30	12	

CC : Contrôle Continu / **CE** : Contrôle Ecrit / **CO** : Contrôle Oral / **CCTP** : Contrôle Continu de TP / **CITP** : Contrôle Individuel de TP

SEMESTRE 8 – DOMINANTE INTELLIGENCE ARTIFICIELLE & BIGDATA

MATHÉMATIQUES POUR L'ANALYSE DE DONNÉES

❖ MATHÉMATIQUES ET STATISTIQUES APPLIQUÉES À L'IA

OBJECTIFS D'APPRENTISSAGE :

1. **Comprendre et appliquer diverses distributions de probabilités**
 - Maîtriser les propriétés et les utilisations des distributions discrètes (Uniforme, Bernoulli, Binomiale, Géométrique, Poisson) et continues (Uniforme, Exponentielle, Normale).
 - Savoir appliquer ces distributions à des problèmes concrets, en lien avec des scénarios réalistes et des applications en intelligence artificielle.
2. **Effectuer des analyses d'inférence statistique**
 - Acquérir les compétences nécessaires pour estimer des paramètres à partir de données échantillonnées et pour tester des hypothèses statistiques.
 - Comprendre et appliquer l'analyse de variance (ANOVA) pour comparer plusieurs groupes et déterminer l'existence de différences statistiquement significatives.
3. **Maîtriser les techniques d'optimisation**
 - Développer une compréhension de l'optimisation sans contrainte pour une et deux variables, en utilisant des méthodes numériques et analytiques adaptées.
 - Apprendre et appliquer des techniques d'optimisation avec contraintes, essentielles pour résoudre des problèmes plus complexes en intelligence artificielle.
4. **Intégration des mathématiques statistiques dans des outils logiciels**
 - Utiliser efficacement R Studio pour le traitement des données, la manipulation des fichiers et des variables, dans le cadre d'analyses statistiques.
 - Améliorer la capacité à interpréter des résultats statistiques et à les représenter graphiquement pour une meilleure prise de décision dans des projets d'IA.

ACQUIS PRÉALABLES :

- Statistiques descriptives :
 - Définitions
 - Représentations graphiques
 - Mesures de position et de dispersion : principes et pratique
 - R studio, manipulation des fichiers et des variables
- Algèbre linéaire (systèmes d'équations linéaires)

PROGRAMME :

- Distributions de probabilités, avec un focus sur les applications pratiques des concepts :
 - Uniforme discrète
 - Bernoulli

- Binomiale
- Géométrique
- Poisson
- Uniforme continue
- Exponentielle
- Normale
- Inférence statistique :
 - Estimation
 - Tests d'hypothèses
 - Analyse de variance (ANOVA)
- Optimisation :
 - Optimisation sans contrainte en une et deux variables
 - Optimisation avec contraintes

❖ MODÉLISATION MATHÉMATIQUE POUR L'IA

OBJECTIFS D'APPRENTISSAGE :

- Maîtriser la régression linéaire pour la modélisation de données
 - Comprendre et appliquer la régression linéaire pour analyser et prédire les relations entre variables.
 - Savoir interpréter les coefficients de régression et les statistiques de diagnostic pour évaluer la qualité des modèles.
- Analyser des séries temporelles à l'aide de modèles ARIMA
 - Acquérir les compétences nécessaires pour modéliser et prédire des données temporelles en utilisant les modèles ARIMA.
 - Apprendre à identifier les composantes saisonnières et non saisonnières dans les séries temporelles et à ajuster les modèles pour des prédictions précises.
- Développer une expertise en programmation mathématique
 - Comprendre et appliquer différentes formes de programmation mathématique, y compris linéaire, entière, non-linéaire et multiobjectif, pour résoudre des problèmes d'optimisation complexes.
 - Savoir formuler des problèmes, choisir et implémenter la méthode d'optimisation appropriée.
- Utiliser des heuristiques pour résoudre des problèmes d'optimisation
 - Maîtriser des techniques heuristiques telles que Glouton, GRASP, et algorithmes génétiques pour trouver des solutions approchées mais efficaces à des problèmes d'optimisation difficiles.
 - Comprendre quand et pourquoi utiliser des heuristiques spécifiques en fonction de la nature du problème et des contraintes de temps ou de ressources.

PROGRAMME :

1. Régression linéaire
2. Séries temporelles (modèles ARIMA)

3. Programmation mathématique (linéaire, entière, non-linéaire, multiobjectif)
4. Heuristiques (Glouton, GRASP, Génétiques, ...)

INTELLIGENCE ARTIFICIELLE

❖ MACHINE LEARNING ET DEEP LEARNING

OBJECTIFS D'APPRENTISSAGE :

1. Acquérir des compétences en programmation Python spécifiques au machine learning, couvrant les bases de Python, les bibliothèques de science des données, et la préparation des données.
2. Comprendre et mettre en œuvre des réseaux de neurones, en apprenant les fondamentaux, l'entraînement et l'optimisation, ainsi que la gestion du surapprentissage.
3. Appliquer le machine learning au traitement du langage naturel, en explorant le prétraitement de texte, les techniques d'extraction de caractéristiques, les plongements de mots et les réseaux neuronaux récurrents.

PROGRAMME :

1. Python pour le Machine Learning
 - Python Basics : syntaxe de base, structures de données.
 - Data Science Libraries : NumPy, pandas, Matplotlib.
 - Data Preprocessing : nettoyage des données, normalisation, codage.
 - Machine Learning Libraries : scikit-learn, TensorFlow, Keras.
2. Réseaux Neuronaux
 - Fondamentaux : architecture des réseaux, propagation avant, rétropropagation.
 - Entraînement et Optimisation : fonctions de coût, descente de gradient, ajustements hyperparamètres.
 - Overfitting : techniques de régularisation, validation croisée.
 - Mise en œuvre d'un MLP (Perceptron Multi-Couches) : construction, entraînement, évaluation.
3. Traitement du Langage Naturel
 - Text Preprocessing : tokenisation, lemmatisation, stop words.
 - Feature Extraction Techniques : sac de mots, TF-IDF.
 - Word Embeddings : Word2Vec, GloVe.
 - Réseaux Neuraux Récurrents : LSTM, GRU pour le traitement de séquences textuelles.

❖ PROJET IA

OBJECTIFS D'APPRENTISSAGE :

- Comprendre les principes fondamentaux des modèles de langage de grande taille (LLM) et leur application.
- Développer des compétences pratiques en manipulation et entraînement de modèles LLM pour la résolution de problèmes spécifiques.
- Participer activement à une compétition Kaggle pour appliquer les connaissances acquises en intelligence artificielle.
- Analyser les résultats et améliorer les performances des modèles en utilisant des techniques avancées d'apprentissage automatique.
- Travailler en équipe pour développer, tester et déployer des solutions IA efficaces.

PROGRAMME :

- Introduction aux Modèles LLM
 - Présentation des concepts de base des modèles de langage de grande taille.
 - Exemples de modèles LLM couramment utilisés (GPT, BERT, etc.).
 - Applications et cas d'usage des LLM.
- Participation à une Compétition Kaggle
 - Introduction aux compétitions Kaggle et exploration des plateformes.
 - Sélection d'une compétition appropriée pour le projet.
 - Analyse des données fournies par la compétition et préparation des données pour l'entraînement.
- Prétraitement des Données
 - Nettoyage des données.
 - Utilisation d'outils et bibliothèques de prétraitement.
- Choix et configuration des modèles LLM.
 - Évaluation des modèles sur les données de la compétition.
 - Méthodes d'évaluation des performances des modèles.
 - Techniques pour améliorer les performances (fine-tuning, ajustement des hyperparamètres, etc.).
 - Analyse des résultats et itération sur le modèle.
- Déploiement des Modèles
 - Introduction aux techniques de déploiement de modèles IA.
 - Utilisation de plateformes cloud pour le déploiement.
 - Mise en place de pipelines CI/CD pour les modèles LLM.
- Travail Collaboratif et Gestion de Projet
 - Outils de gestion de projet et de collaboration (Git, Jira, etc.).
 - Techniques de travail en équipe pour le développement de solutions IA.
 - Documentation et présentation des résultats.
 - Soumission et Participation Active à la Compétition
- Préparation des soumissions pour Kaggle.
 - Stratégies pour maximiser les scores et les performances.

- Participation active à la communauté Kaggle et interaction avec d'autres participants.

BIG DATA

❖ ARCHITECTURE BIG DATA ET CLOUD COMPUTING

PRÉ-REQUIS :

- Connaissances d'une base de données relationnelles (Oracle, SQL Server, PostgreSQL, ...)
- Familiarité avec les environnements Linux (Bash / WSL)
- Connaissances du langage Python
- Utilisation d'un IDE (VS Code)

OBJECTIFS :

- Définir les principes fondamentaux du Big Data
- Les différences fondamentales par rapport aux bases de données relationnelles traditionnelles
- Utiliser un environnement de développement permettant d'utiliser une infrastructure Big Data afin pouvoir écrire, exécuter, analyser et optimiser des traitements de données (calculs d'agrégats et transformations)

ACQUIS D'APPRENTISSAGE VISÉS :

- Reconnaître et utiliser le vocabulaire afférent au domaine du Big Data et du Cloud Computing
- Comprendre et décrire les différents concepts architecturaux du Big Data
- Analyser et évaluer une architecture Big Data
- Décrire les enjeux liés Big Data
- Appréhender les contraintes liées aux traitements de données massives et distribué
- Concevoir et implémenter une chaîne de traitement sur un environnement Big Data dans le Cloud

PROGRAMME :

❖ BASES DE DONNÉES BIG DATA AVEC NOSQL

OBJECTIFS :

- Comprendre les principes fondamentaux des bases de données NoSQL, leurs types, et leurs utilisations.
- Concevoir et implémenter des solutions de bases de données NoSQL en utilisant MongoDB pour les bases de données de documents et Neo4j pour les bases de données graphes.
- Utiliser des outils de développement et d'intégration continue pour gérer et déployer des applications utilisant des bases de données NoSQL.
- Déployer des bases de données NoSQL dans un environnement cloud en utilisant GCP.

ACQUIS D'APPRENTISSAGE VISÉS :

- Concevoir et réaliser des opérations CRUD avec MongoDB et Neo4j
- Modéliser et interagir avec des bases de données de type document et graphes

- Interagir avec des bases de données MongoDB et Neo4j en utilisant des drivers Python (PyMongo, Neo4j, etc...)
- Comprendre et appliquer les concepts de BASE vs ACID, Théorème CAP.
- Conteneuriser et déployer des bases de données NoSQL sur des serveurs Cloud
- Utiliser Git pour collaborer sur un projet
- Utiliser GitHub Actions pour l'intégration continue (automatisation des tests unitaires et d'intégration)
- Intégrer les bonnes pratiques de code pour écrire du code visant à interagir avec les bases de données

PROGRAMME :

- Introduction au NoSQL
- Principes de Conception de Bases de Données NoSQL
- Introduction aux Bases de Données Documents et Graphes
- Modélisation des données avec Mongo et Neo4j
- Déploiement de bases de données NoSQL via IaC
- Projet fil rouge

❖ **GOOGLE CLOUD : ASSOCIATE CLOUD ENGINEER**

Ce cours est attaché à la certification [Associate Cloud Engineer](#)

OBJECTIFS D'APPRENTISSAGE :

Les Associate Cloud Engineers déploient des applications, surveillent des opérations et gèrent des solutions d'entreprise. Ils utilisent Google Cloud Console et l'interface de ligne de commande pour effectuer des tâches courantes basées sur la plate-forme afin de maintenir une ou plusieurs solutions déployées qui exploitent les services autogérés ou gérés par Google sur Google Cloud.

L'examen Associate Cloud Engineer évalue les compétences suivantes :

- Configuration d'un environnement de solution cloud
- Planification et configuration d'une solution cloud
- Déploiement et mise en œuvre d'une solution cloud
- Garantie du bon fonctionnement d'une solution cloud
- Configuration des accès et de la sécurité

PROGRAMME :

Section 1 : Configurer un environnement de solution cloud

- 1.1 Configurer des projets cloud et des comptes
- 1.2 Gérer la configuration de la facturation
- 1.3 Installer et configurer l'interface de ligne de commande (CLI), particulièrement le SDK Cloud (ex. : configurer le projet par défaut)

Section 2 : Planifier et configurer une solution cloud

- 2.1 Planifier l'utilisation des produits Google Cloud et estimer son coût à l'aide du calculateur de prix
- 2.2 Planifier et configurer les ressources de calcul
- 2.3 Planifier et configurer les options de stockage de données
- 2.4 Planifier et configurer des ressources réseau

Section 3 : Déployer et mettre en œuvre une solution cloud

- 3.1 Déployer et mettre en œuvre des ressources Compute Engine
- 3.2 Déployer et mettre en œuvre des ressources Google Kubernetes Engine
- 3.3 Déployer et mettre en œuvre des ressources Cloud Run et Cloud Functions
- 3.4 Déployer et mettre en œuvre des solutions de données
- 3.5 Déployer et mettre en œuvre des ressources réseau
- 3.6 Déployer une solution à l'aide de Cloud Marketplace
- 3.7 Mettre en œuvre des ressources via une Infrastructure as Code (IaC)

Section 4 : Garantir le bon fonctionnement d'une solution cloud

- 4.1 Gérer les ressources Compute Engine
- 4.2 Gérer les ressources Google Kubernetes Engine
- 4.3 Gérer les ressources Cloud Run
- 4.4 Gérer les solutions de stockage et de base de données
- 4.5 Gérer les ressources réseau
- 4.6 Gérer la surveillance et la journalisation

Section 5 : Configurer les accès et la sécurité

- 5.1 Utiliser la gestion de l'authentification et des accès (IAM)
- 5.2 Gérer les comptes de service
- 5.3 Afficher les journaux d'audit.

MASSSES HORAIRES, COEFFICIENTS ET
CONTROLES
3ème année Semestres 9 et 10
FILIERE INITIALE STATUT APPRENTI
ANNÉE 2024/2025

(*) Les étudiants des dominantes IA et IF ne sont pas concernés par les Approches-Métiers et les Electifs. Pour les étudiants de ces dominantes, l'unité d'enseignement ENSEIGNEMENTS TRANSVERSAUX compte pour 2 crédits ECTS.

UE	Réf.	Module	SEMESTRE	COURS	TD	TP	TOTAL	COEF.	ECTS*	CONTROLES
ENSEIGNEMENTS TRANSVERSAUX	ET3C1-F	Approches Métiers (*)	9	30			30	6		CC
	ET3C2-A	Anglais	9		30		30	6	5	Groupes préparation au TOEIC : une note finale (composée de CC1 40%, CC2 40%, CC3 Production/Participation 20%) ou score TOEIC converti sur 20 (785 pts = 10/20). Groupes anglais thématique : une note finale de CC ou score TOEIC converti sur 20 (785 points = 10/20)
	ET3C3-F	Electif (**)	9	18			18	6		CC
	ET3C4-F	Simulation de recrutement	9	1	1		2			
PROJET	PI3C1-F	Projet ingénieur	9			150	150	30	10	CC Binôme/Soutenance
MISSION ENTREPRISE	MI3C1-F	Mission industrielle 3ème année	10						27	Rapp. / Sout. + Evaluation pratique par l'Entreprise
PROJET PERSONNEL ET PROFESSIONNEL	PP3C1-F	Projet Personnel et Professionnel	10						3	Rapport
				49	31	150	230	48	45	
DOMINANTES										
UE	Réf.	Module	SEMESTRE	COURS	TD	TP	TOTAL	COEF.	ECTS	
Dominantes		Dominantes	9				220	42	15	
							450	90	60	

CC : Contrôle Continu / CE : Contrôle Ecrit / CO : Contrôle Oral / CCTP : Contrôle Continu de TP / CITP : Contrôle Individuel de TP

(**) Un électif est à choisir parmi les suivants :
- Management des organisations
- Management de situation de crise

MASSSES HORAIRES, COEFFICIENTS ET CONTROLES
2ème année Dominante Développement Logiciel : Test & Qualité (DLTQ)
Année 2024/2025

UE	Réf.	Module	Semestre	COURS	TD	TP	EXAMENS	TOTAL	COEF. MOD	ECTS	CONTROLES
Analyse et conception	DLTQ201-A	Algorithmique et structures de données	8	10	6	12	2	30	5	6	1 CC/1 CE
	DLTQ202-A	Génie Logiciel : UML et Design Pattern	8	10	6	12	2	30	5		1 CCTP/1 CE/1 CTP
	DLTQ203-A	Bases de données : SQL et NoSQL	8	8	4	16	2	30	5		1 CCTP/1 CE/1 CTP
Développement logiciel, Architecture & Test	DLTQ204-F	Développement logiciel et DevOps	8	8	0	20	2	30	5	6	1 CC/1 CE
	DLTQ205-F	Programmation orientée objet	8	8	0	20	2	30	5		1 CC/1 CE
	DLTQ206-A	TDD et Qualité du code	8	8	0	20	2	30	5		1 CC/1 CE
				52	16	100	12	180	30	12	

CC : Contrôle Continu / **CE** : Contrôle Ecrit / **CO** : Contrôle Oral / **CCTP** : Contrôle Continu de TP / **CITP** : Contrôle Individuel de TP

MASSES HORAIRES, COEFFICIENTS ET CONTROLES
3ème année Dominante Intelligence Artificielle et Big Data (IA&BD)
Année 2024/2025

UE	Réf.	Module	SEMESTRE	COURS	TD	TP	EXAMENS	TOTAL	COEF. MOD	ECTS	CONTROLES
Intelligence Artificielle & Big Data	IABD-301-A	Intelligence Artificielle avancée	8	12	6	20	2	40	8		1 CC/1 CE
	IABD-302-F	Architecture Big Data et Intelligence Artificielle	8	8	0	16	1	25	5		1 CC/1 CE
	IABD-303-F	Déploiement de solution IA et Cybersécurité	8	8	0	20	2	30	6	8	1 CC/1 CE
	IABD-304-A	Google Cloud : Professionnel Cloud Architect	8	12	0	12	1	25	5		1 CC (conversion du score de validation selon les critères d'obtention de la certification) / 1 CE
Data Gouvernance, Green IA et Approche Responsable de l'IA	IABD-305-F	Green IA & Approche responsable	8	10	0	8	2	20	3	3	1 CC/1 CE
	IABD-306-F	Data Gouvernance & RGPD	8	8	0	20	2	30	6		1 CC/1 CE
Projet IA & Big Data	IABD-307-F	Projet IA & Big Data : de la conception au déploiement	8	6	0	44		50	9	4	1 CC
				64	6	140	10	220	42	15	

CC : Contrôle Continu / **CE** : Contrôle Ecrit / **CO** : Contrôle Oral / **CCTP** : Contrôle Continu de TP / **CITP** : Contrôle Individuel de TP

SEMESTRE 9 – DOMINANTE DÉVELOPPEMENT LOGICIEL - TEST & QUALITÉ

QUALITÉ LOGICIELLE

❖ ASSURANCE QUALITÉ

OBJECTIFS D'APPRENTISSAGE :

- Comprendre les fondamentaux et les enjeux de la qualité logicielle et de l'assurance qualité dans les projets de développement.
- Savoir élaborer et mettre en œuvre des stratégies de tests et de validation permettant d'assurer la qualité des logiciels.
- Apprendre à utiliser et à intégrer des outils et des méthodes modernes pour l'automatisation des tests et la gestion de la qualité.
- Développer une capacité à analyser et à améliorer les processus de développement logiciel pour augmenter leur efficacité et leur qualité.
- Préparer la certification ISTQB

PROGRAMME :

- **Fondamentaux de la qualité logicielle**
 - Introduction aux concepts de qualité logicielle.
 - Importance de la qualité pour la sécurité et la maintenance des logiciels.
- **Normes et réglementations de qualité logicielle**
 - Présentation des principales normes : ISO 9001, CMMI, ITIL, ECSS.
 - Introduction à l'analyse statique (e.g : MISRA C++): objectifs, principes, et rôle dans la prévention des erreurs de programmation.
- **Processus d'assurance qualité logicielle**
 - Définition des rôles et des responsabilités dans l'assurance qualité.
 - Planification et mise en œuvre de l'assurance qualité dans les cycles de développement.
- **Outils d'assurance qualité et de test**
 - Utilisation d'outils comme SONAR, JENKINS, TESTLINK pour le contrôle et l'amélioration de la qualité.
 - Test Driven Development (TDD) et automatisation des tests : outils et techniques.
- **Prévention des risques et gestion de la qualité**
 - Techniques d'identification et d'analyse des risques qualité.
 - Stratégies pour la mitigation des risques et l'amélioration continue.
- **Intégration de l'analyse statique dans les projets logiciels**
 - Application concrète de directives normatives dans des exemples de code.
 - Cas pratiques sur l'adoption de normes sans compromettre la performance et la fonctionnalité du logiciel.
- **Éthique et responsabilité en qualité logicielle**
 - Discussions sur les implications éthiques des pratiques de qualité.
 - Importance du Green IT et des considérations environnementales dans le développement logiciel.

❖ PERFORMANCE ET STABILITÉ DES APPLICATIFS

OBJECTIFS D'APPRENTISSAGE :

- Comprendre les principes fondamentaux de la performance et de la stabilité des applications logicielles.
- Identifier et résoudre les problèmes affectant la performance et la stabilité des applicatifs.
- Maîtriser les outils et techniques pour mesurer, évaluer et améliorer la performance des applications.
- Appliquer les meilleures pratiques de développement pour garantir la stabilité et la performance lors de la conception des applications.

PROGRAMME

- **Introduction à la performance des applications**
 - Définitions de base : performance, réactivité, efficacité.
 - Importance de la performance dans le cycle de vie du développement logiciel.
- **Analyse de la performance des applications**
 - Techniques de mesure de la performance : profiling, benchmarking.
 - Utilisation d'outils de monitoring pour suivre la performance en temps réel.
- **Optimisation de la performance**
 - Techniques d'optimisation au niveau du code : algorithmique, structures de données adaptées.
 - Optimisation des interactions avec les bases de données : requêtes, indexation.
- **Stabilité des applications**
 - Définition et importance de la stabilité dans les applications logicielles.
 - Gestion des erreurs et exceptions pour maintenir la stabilité.
- **Tests de charge et de stress**
 - Planification et exécution de tests de charge pour simuler des conditions d'utilisation réelle.
 - Analyse des résultats de tests de stress pour identifier les points de rupture.
- **Maintien de la performance et de la stabilité**
 - Stratégies de maintenance proactive pour préserver la performance et la stabilité.
 - Étude de cas sur l'impact de la performance et de la stabilité sur l'expérience utilisateur.

❖ DÉVELOPPEMENT LOGICIEL ET SÉCURITÉ

OBJECTIFS D'APPRENTISSAGE :

- Comprendre les principes fondamentaux et les meilleures pratiques de sécurité dans le développement logiciel.
- Maîtriser les outils et techniques de programmation sécurisée.
- Savoir concevoir, tester, et déployer des applications robustes et sécurisées.
- Développer une capacité à évaluer et à améliorer la sécurité des applications existantes.
- Favoriser une culture de sécurité informatique parmi les futurs ingénieurs.

PROGRAMME :

- Introduction à la sécurité informatique
 - Concepts de base de la sécurité des systèmes d'information.
 - Menaces, vulnérabilités, et mécanismes de défense.
- Programmation sécurisée
 - Techniques de codage sécurisé pour prévenir les vulnérabilités courantes (injections SQL, XSS, etc.).
 - Utilisation de bibliothèques et frameworks sécurisés.
- Authentification et gestion des sessions
 - Protocoles d'authentification.
 - Sécurisation des sessions et des communications.
- Tests de sécurité
 - Tests d'intrusion (pentesting).
 - Tests automatisés pour la sécurité (SAST, DAST).
- Déploiement sécurisé
 - Sécurité dans le cycle de développement (DevSecOps).
 - Chiffrement et gestion des clés dans les environnements de production.
- Cas pratiques et projets
 - Études de cas sur les incidents de sécurité et leurs résolutions.

❖ ARCHITECTURE DES SI ET CYBERSÉCURITÉ

OBJECTIFS D'APPRENTISSAGE :

- Comprendre les principes fondamentaux de l'architecture des systèmes d'information (SI) : Apprendre à concevoir et à analyser les structures des systèmes d'information pour répondre efficacement aux besoins organisationnels.
- Maîtriser les aspects clés de la cybersécurité : Développer une compréhension approfondie des menaces, des vulnérabilités, et des mesures de protection pour sécuriser les systèmes d'information.
- Intégrer des solutions de cybersécurité dans l'architecture des SI : Savoir comment incorporer des pratiques de sécurité dès la phase de conception et tout au long du cycle de vie des systèmes d'information.
- Analyser et gérer les risques liés à la sécurité informatique : Être capable d'identifier, d'évaluer et de gérer les risques associés aux informations et aux systèmes informatiques.

PROGRAMME :

- Introduction à l'architecture des systèmes d'information
 - Définition et composants clés d'un système d'information.
 - Modèles d'architecture des SI (par ex., architectures n-tiers, microservices).
- Fondamentaux de la cybersécurité
 - Principes de base de la cybersécurité : confidentialité, intégrité, disponibilité.
 - Types de menaces et mécanismes de défense (malware, phishing, attaques DDoS).
- Sécurité des réseaux et des communications
 - Sécurisation des réseaux d'entreprise.
 - Protocoles sécurisés (SSL/TLS, VPN, IPSec).
- Gestion des identités et des accès (IAM)
 - Principes et solutions pour la gestion des identités et des accès.
 - Authentification forte, autorisation et gestion des privilèges.
- Sécurité des applications
 - Sécurité dans le développement des logiciels.
 - Prévention des vulnérabilités communes (injections SQL, XSS).
- Gestion des incidents et plan de réponse aux incidents
 - Établissement d'un plan de réponse aux incidents de sécurité.
 - Simulations et exercices de gestion des incidents.
- Aspects légaux et conformité en cybersécurité
 - Compréhension des exigences réglementaires (GDPR, lois nationales sur la cybersécurité).
 - Implications de la non-conformité et gestion des audits de sécurité.

❖ ADMINISTRATION SYSTÈME & INFRASTRUCTURE AS CODE

OBJECTIFS D'APPRENTISSAGE :

- Comprendre et appliquer les principes de la programmation système pour gérer et optimiser les interactions avec la couche matérielle (hardware).
- Maîtriser l'Infrastructure as Code (IaC) pour automatiser la configuration et la gestion des infrastructures virtuelles et physiques.
- Savoir améliorer l'efficacité des processus de déploiement en écrivant des scripts et utilisant des outils d'automatisation.
- Acquérir une compréhension approfondie des systèmes d'exploitation et de leur fonctionnement interne.
- Savoir utiliser des plateformes de gestion de configuration telles que Puppet, Chef, et Ansible.

PROGRAMME :

- Introduction à la programmation système
 - Concepts de base des systèmes d'exploitation
 - Gestion des processus et threads
 - Manipulation des fichiers et des systèmes de fichiers
 - Communication entre processus
- Infrastructure as Code (IaC)
 - Principes et avantages de l'IaC
 - Outils et plateformes d'IaC (Terraform, Ansible, Chef)
 - Écriture de scripts d'automatisation pour le déploiement et la gestion des infrastructures
 - Utilisation de conteneurs et de systèmes d'orchestration comme Docker et Kubernetes
- Programmation pour la performance et la sécurité
 - Optimisation de code pour l'amélioration des performances système
 - Principes de sécurisation des systèmes à travers la programmation
 - Gestion des droits d'accès et de la sécurité des processus
- Projets et études de cas
 - Développement de scripts pour l'automatisation de tâches système
 - Mise en place d'une infrastructure virtuelle avec IaC pour un cas d'usage spécifique
 - Analyse et amélioration de la sécurité et de la performance d'un système existant

❖ DÉPLOIEMENT ET CLOUD COMPUTING

OBJECTIFS D'APPRENTISSAGE :

- Comprendre les fondamentaux du Cloud Computing : Appréhender les différents modèles de service comme IaaS, PaaS, SaaS et les modèles de déploiement : public, privé, hybride, et communautaire.
- Maîtriser les technologies de virtualisation : Savoir comment les technologies de virtualisation sous-tendent la plupart des offres de cloud computing.
- Développer des compétences en gestion et automatisation du cloud : Utiliser des outils tels que Kubernetes et Docker pour la gestion de conteneurs, et des outils d'automatisation comme Ansible, Chef, ou Puppet.
- Sécurité dans le cloud : Comprendre les enjeux de sécurité spécifiques au cloud et apprendre à mettre en œuvre des pratiques de sécurité robustes.
- Conception et déploiement d'applications scalables : Apprendre à concevoir, tester et déployer des applications scalables et résilientes sur le cloud.

PROGRAMME :

- Introduction au Cloud Computing
 - Cf Certification CDL
- Technologies de virtualisation
 - Hyperviseurs et gestion de VM
 - Conteneurisation et orchestration avec Docker et Kubernetes
- Gestion et automatisation du Cloud
 - Introduction aux outils d'automatisation : Ansible, Chef, Puppet
 - Automatisation des déploiements et gestion de la configuration
- Sécurité dans le Cloud
 - Meilleures pratiques de sécurité dans le cloud
 - Gestion des identités et des accès
 - Protection des données et conformité réglementaire
- Développement et déploiement d'applications dans le cloud
 - Architectures sans serveur et microservices
 - CI/CD pipelines pour le cloud
 - Monitoring et optimisation des performances
- Cas pratiques et projets
 - Mise en place d'une architecture microservices sur une plateforme cloud
 - Simulation d'une migration d'infrastructure IT vers un modèle cloud hybride
 - Projets de déploiement d'applications utilisant des services cloud spécifiques (AWS, Azure, Google Cloud)

OBJECTIFS D'APPRENTISSAGE :

- Concevoir et développer une application client-serveur intégrant un client Desktop (C++ avec Qt) et un tableau de bord web pour la visualisation des données.
- Appliquer les principes de gestion de projet pour planifier, suivre, et livrer le projet selon les critères de qualité.
- Utiliser les bonnes pratiques de développement, y compris le développement orienté objet, les tests unitaires, et l'intégration continue.
- Mettre en œuvre une architecture serveur robuste et sécurisée pour gérer les communications et le stockage des données.
- Intégrer des outils de CI/CD pour l'automatisation des tests et du déploiement.
- Développer des compétences en travail d'équipe, communication, et gestion du temps.

PROGRAMME :

- **Conception et Modélisation**
 - Utilisation de Design Patterns pour résoudre des problèmes récurrents en développement logiciel.
 - Architecture du système : conception de l'architecture client-serveur, spécification des interfaces et protocoles de communication.
- **Gestion de Projet et Méthodes Agiles**
 - Planification, estimation et répartition des tâches, gestion des risques.
 - Utilisation des outils de gestion de projet (Jira, Trello).
 - Méthodologies Agiles : SCRUM, Kanban.
- **Développement du Client Desktop**
 - Introduction à Qt pour le développement d'interfaces graphiques en C++.
 - Développement d'une application desktop avec Qt : gestion des événements, création de widgets, communication réseau.
 - Intégration des données provenant du serveur dans l'interface utilisateur.
- **Développement du Tableau de Bord Web**
 - Développement d'un tableau de bord interactif pour visualiser les données.
 - Sécurisation de l'accès au tableau de bord.
- **Développement de la Partie Serveur**
 - Conception et implémentation de l'API serveur (RESTful ou GraphQL).
 - Utilisation de bases de données SQL et/ou NoSQL pour le stockage des données.
 - Sécurisation des communications serveur-client (SSL/TLS).
 - Gestion des sessions et authentification des utilisateurs.
- **Tests et Assurance Qualité**
 - Développement de tests unitaires et fonctionnels.
 - Utilisation des frameworks de tests (Google Test pour C++, Jest pour JavaScript).
 - Utilisation Test Driven Development (TDD).
 - Mise en place de l'intégration continue (CI) et du déploiement continu (CD) avec des outils comme Jenkins, GitLab CI, ou GitHub Actions.
- **Déploiement et DevOps**
 - Mise en place et gestion des environnements de développement, de test et de production.
 - Utilisation d'outils de contrôle de version (Git).
 - Surveillance et logging des applications déployées.

- **Projet Final**

- Conception, développement et déploiement d'une application complète intégrant toutes les composantes vues en cours.
- Documentation du projet : spécifications techniques, guide utilisateur, et manuel de maintenance.
- Présentation et démonstration du projet devant un jury.

SEMESTRE 9 – DOMINANTE INTELLIGENCE ARTIFICIELLE & BIGDATA

INTELLIGENCE ARTIFICIELLE ET BIG DATA

❖ INTELLIGENCE ARTIFICIELLE AVANCÉE

OBJECTIFS D'APPRENTISSAGE :

- **Comprendre l'architecture des LLM** : Découvrir les concepts fondamentaux de l'architecture Transformer, spécifiquement le modèle GPT utilisé dans les LLM modernes.
- **Maîtriser les mécanismes d'attention** : Appréhender les principes des mécanismes d'attention, tels que l'attention à produit scalaire normalisée, essentiels pour le traitement des données textuelles.
- **Générer des textes avec des stratégies avancées** : Explorer différentes méthodes de génération de texte comme le décodage gourmand, la recherche en faisceau et l'échantillonnage top-k.
- **Création et amélioration de datasets d'instruction** : Apprendre à générer des données synthétiques de haute qualité pour le fine-tuning des LLM et à améliorer les datasets existants avec des techniques avancées.
- **Fine-tuning supervisé des modèles pré-entraînés** : Comprendre et appliquer le fine-tuning supervisé pour adapter les modèles pré-entraînés à des tâches spécifiques.
- **Utilisation du Reinforcement Learning from Human Feedback (RLHF)** : Intégrer le RLHF pour aligner les réponses des modèles avec les attentes humaines et réduire les biais.
- **Évaluation des LLM** : Développer une compréhension des différentes méthodes et métriques d'évaluation des modèles de langage, y compris les évaluations humaines et les benchmarks spécifiques à des tâches.
- **Quantification des modèles pour l'efficacité** : Se familiariser avec les techniques de quantification qui réduisent les coûts computationnels et de mémoire des modèles.
- **Tendances récentes dans les LLM** : Explorer les nouvelles avancées comme les encodages positionnels relatifs, la fusion de modèles, et les architectures Mixture of Experts.

PROGRAMME :

1. **Architecture LLM**
 - Vue d'ensemble de l'architecture Transformer
 - Compréhension du modèle GPT
2. **Mécanismes d'attention**
 - Théorie de l'attention et attention à produit scalaire
3. **Génération de texte**
 - Stratégies de génération de texte : décodage gourmand, recherche en faisceau
4. **Dataset d'instruction**
 - Création de datasets synthétiques
 - Amélioration de datasets via techniques avancées
5. **Fine-tuning supervisé**
 - Techniques de fine-tuning pour l'adaptation aux tâches spécifiques
6. **RLHF**
 - Intégration du feedback humain pour améliorer la pertinence des réponses
7. **Évaluation des modèles**

- Métriques traditionnelles et benchmarks spécifiques
- Évaluations humaines
- 8. **Quantification des modèles**
 - Techniques de réduction de la précision pour l'efficacité
- 9. **Nouvelles tendances**
 - Encodages positionnels relatifs et architectures Mixture of Experts

❖ ARCHITECTURE BIG DATA APPLIQUÉE À INTELLIGENCE ARTIFICIELLE

OBJECTIFS D'APPRENTISSAGE :

- Comprendre les concepts fondamentaux des data lakes et des data warehouses.
- Construire des batch data pipelines pour le traitement des données volumineuses.
- Développer des systèmes d'analyse de streaming résilients.
- Appliquer des techniques de smart analytics, de machine learning et d'IA sur des plateformes cloud.
- Maîtriser les fondations du traitement des données serverless.
- Développer et gérer des pipelines de traitement des données serverless.
- Préparer les données pour l'utilisation des API de machine learning.
- Concevoir et construire un data warehouse efficace.
- Implémenter l'ingénierie des données pour la modélisation prédictive.
- Concevoir et développer une architecture de data mesh.

PROGRAMME :

- **Batch Data Pipelines**
 - Conception et construction de pipelines de traitement par lots
 - Outils et technologies (ex: Apache Hadoop, Apache Spark)
 - Optimisation et gestion des performances
- **Systèmes d'Analyse de Streaming**
 - Concepts et architecture des systèmes de streaming
 - Outils et technologies (ex: Apache Kafka, Apache Flink)
 - Développement de solutions résilientes pour le streaming de données
- **Smart Analytics, Machine Learning et IA sur le Cloud**
 - Introduction aux services cloud pour l'IA et l'analytique
 - Utilisation de plateformes cloud (ex: Google Cloud, AWS, Azure)
 - Déploiement et gestion de modèles de machine learning sur le cloud
- **Traitement des Données Serverless : Fondations**
 - Concepts de serverless computing
 - Services serverless (ex: AWS Lambda, Google Cloud Functions)
 - Avantages et défis des architectures serverless
- **Traitement des Données Serverless : Développer des Pipelines**
 - Conception de pipelines serverless
 - Intégration de services serverless pour le traitement des données
 - Bonnes pratiques et optimisation des performances
- **Traitement des Données Serverless : Opérations**
 - Surveillance et gestion des pipelines serverless

- Sécurité et gouvernance des données dans un environnement serverless
- Automatisation et scaling des opérations
- **Préparation des Données pour les ML APIs**
 - Techniques de nettoyage et de transformation des données
 - Intégration des données pour les modèles de machine learning
 - Utilisation des API de machine learning pour les prédictions
- **Construire un Data Warehouse**
 - Conception et modélisation d'un data warehouse
 - Sélection des technologies et outils (ex: Snowflake, Amazon Redshift)
 - Stratégies de chargement et de transformation des données (ETL/ELT)
- **Ingénierie des Données pour la Modélisation Prédictive**
 - Processus de préparation des données pour la modélisation prédictive
 - Outils et techniques pour l'ingénierie des fonctionnalités
 - Évaluation et validation des modèles prédictifs
- **Construire un Data Mesh**
 - Concepts et principes de l'architecture de data mesh
 - Décentralisation et gouvernance des données
 - Cas d'utilisation et exemples d'implémentation

❖ DÉPLOIEMENT DE SOLUTION IA ET CYBERSÉCURITÉ

OBJECTIFS D'APPRENTISSAGE :

- Acquérir les compétences nécessaires pour déployer des solutions d'IA sur des plateformes cloud comme AWS, Google Cloud Platform (GCP) et Microsoft Azure.
- Maîtriser les outils et techniques de Feature Engineering pour améliorer les modèles de machine learning.
- Apprendre à intégrer et utiliser des systèmes de machine learning en production.
- Découvrir les bases du Computer Vision et du Natural Language Processing (NLP) sur le cloud.
- Mettre en place et gérer des opérations de machine learning (MLOps) pour assurer la qualité et l'efficacité des solutions déployées.
- Comprendre les enjeux de la cybersécurité dans le contexte des solutions IA et apprendre à sécuriser les systèmes de ML.
- Développer des compétences en gestion de projets IA et en stratégie d'intégration de la cybersécurité dans les solutions d'IA.
- Apprendre à utiliser des plateformes comme TensorFlow sur le cloud pour développer des applications IA robustes.

PROGRAMME :

- **Développement et déploiement de modèles sur le cloud**
 - Introduction à TensorFlow
 - Développement et déploiement de modèles sur frameworks type TensorFlow sur des plateformes cloud
 - Techniques avancées de Feature Engineering
 - Outils et pipelines pour le Feature Engineering sur le cloud

- **Machine Learning dans l'entreprise**
 - Cas d'utilisation du ML dans différents secteurs d'activité
 - Déploiement de solutions ML à l'échelle de l'entreprise
 - Construction de pipelines de ML efficaces
 - Utilisation des services cloud pour les pipelines de ML
- **Mise en production et opérations de machine learning (MLOps)**
 - Déploiement et monitoring des modèles ML en production
 - Gestion des mises à jour et des performances des modèles
 - Outils et frameworks pour MLOps (ex : MLflow) pour le suivi des modèles et des datasets
- **Applications spécialisées de l'IA sur le cloud**
 - Développement et déploiement de solutions de vision par ordinateur sur le cloud
 - Développement et déploiement de modèles NLP sur le cloud
 - Systèmes de recommandation : Techniques, algorithmes et déploiement sur le cloud
- **Préparation et intégration des données**
 - Préparation et nettoyage des données
 - Utilisation des APIs de ML pour l'intégration des données
- **Déploiement de solutions de machine learning sur des plateformes cloud**
 - Introduction à AWS SageMaker, GCP Vertex AI et Azure ML
 - Étapes de développement, entraînement et déploiement des modèles sur ces plateformes
 - Bonnes pratiques de cybersécurité pour chaque étape du déploiement

❖ GOOGLE CLOUD : PROFESSIONAL CLOUD ARCHITECT

OBJECTIFS D'APPRENTISSAGE :

Les architectes cloud professionnels permettent aux entreprises de tirer parti des technologies Google Cloud. Avec une compréhension approfondie de l'architecture cloud et de Google Cloud, ils conçoivent, développent et gèrent des solutions robustes, sécurisées, évolutives, hautement disponibles et dynamiques pour atteindre les objectifs commerciaux.

PROGRAMME :

L'examen de certification Professional Cloud Architect évalue les compétences suivantes :

- Conception et planification de l'architecture d'une solution cloud
- Gestion et provisionnement de l'infrastructure de la solution cloud
- Conception de solutions sécurisées et conformes
- Analyse et optimisation des processus techniques et métier
- Gestion de la mise en œuvre d'une architecture cloud
- Fiabilité de la solution et des opérations

❖ GREEN IA & APPROCHE RESPONSABLE

OBJECTIFS D'APPRENTISSAGE :

- Comprendre les principes de l'IA responsable et les situations où l'IA n'apporte pas de valeur ajoutée.
- Apprendre choisir des modèles IA petits et spécifiques ou des modèles généralistes et volumineux.
- Connaître les différences et implications entre les modèles IA locaux et ceux hébergés sur le cloud.
- Étudier des exemples concrets comme la synthèse vocale pour illustrer des approches IA efficaces.
- Analyser l'impact sociétal de l'IA, incluant les biais, les aspects juridiques et éthiques.
- Explorer des cas d'usage de l'IA dans différents domaines comme le recrutement, la gestion financière et l'optimisation budgétaire.
- Intégrer une vision FinOps pour gérer les coûts et l'efficacité des solutions IA.

PROGRAMME :

- **Introduction à l'IA responsable**
 - Définitions et concepts de l'IA responsable
 - Cas où l'IA n'apporte pas de valeur ajoutée
- **Green IA**
 - Choix de modèles IA adaptés et spécifiques
 - Différences entre IA locale et cloud
 - Étude de cas avec des modèles généralistes, légers et sans GenAI (ex : Synthèse vocale)
- **Impact sociétal et éthique**
 - Biais dans les algorithmes IA
 - Aspects juridiques et réglementations
 - Éthique et déontologie en IA
- **Applications pratiques et cas d'usage**
 - Recrutement par IA
 - Gestion budgétaire et prises de décision automatisées
 - FinOps : Optimisation financière des solutions IA
- **Vision globale et futur de l'IA**
 - Innovations et tendances futures en IA responsable
 - Pratiques exemplaires et initiatives écologiques dans le domaine de l'IA
 - Conclusion et perspectives sur l'IA durable et responsable

❖ DATA GOUVERNANCE & RGPD

OBJECTIFS D'APPRENTISSAGE

- **Comprendre les principes fondamentaux de la gouvernance des données** : Savoir comment les données sont collectées, stockées, gérées et sécurisées au sein d'une organisation.
- **Maîtriser les aspects réglementaires du RGPD** : Connaître les obligations légales concernant la protection des données personnelles et leur application pratique dans le cadre professionnel.

- **Analyser et évaluer les risques liés à la gestion des données** : Développer des compétences pour identifier et évaluer les risques en matière de confidentialité et de sécurité des données.
- **Implémenter des politiques de conformité RGPD** : Être capable de mettre en place et de maintenir des politiques de conformité adaptées aux normes RGPD.

PROGRAMME

- **Introduction à la gouvernance des données**
 - Définition et importance de la gouvernance des données.
 - Les principaux modèles et meilleures pratiques de gouvernance des données.
- **Principes du RGPD**
 - Aperçu du Règlement Général sur la Protection des Données (RGPD).
 - Droits des individus sous le RGPD et obligations des organisations.
- **Mise en œuvre du RGPD dans les opérations quotidiennes**
 - Étapes pour établir une conformité RGPD dans une organisation.
 - Utilisation d'outils et de technologies pour soutenir la conformité au RGPD.
- **Gestion des incidents et des violations de données**
 - Procédures à suivre en cas de violation de données.
 - Comment documenter et signaler une violation en conformité avec le RGPD.
- **Audit et surveillance de la conformité**
 - Techniques d'audit pour assurer le respect continu du RGPD.
 - Comment évaluer et améliorer les mesures de sécurité des données.

❖ PROJET IA & BIG DATA : DE LA CONCEPTION AU DÉPLOIEMENT

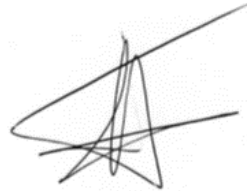
OBJECTIFS D'APPRENTISSAGE :

- **Comprendre le fonctionnement et l'exécution des modèles LLM (Large Language Models) :**
 - Familiarisation avec les API LLM et les modèles open-source.
 - Maîtrise des techniques de prompt engineering et de structuration des sorties.
- **Construire et gérer un stockage vectoriel pour le traitement des données :**
 - Apprendre à charger, diviser, et stocker des documents sous forme de vecteurs pour une utilisation future.
- **Développer des compétences en Retrieval Augmented Generation (RAG) :**
 - Intégrer et manipuler des documents contextuels pour améliorer la précision des réponses des modèles.
- **Optimiser le processus d'inférence pour les applications en production :**
 - Implémenter des techniques avancées comme Flash Attention et Key-value cache pour réduire les coûts et accélérer l'inférence.
- **Déployer les LLM dans différents environnements (local, serveur, edge) :**
 - Évaluer les différentes stratégies de déploiement et les meilleures pratiques pour les applications en production.
- **Sécuriser les applications basées sur LLM contre diverses menaces :**
 - Appliquer des mesures défensives pour protéger les applications des injections de prompt et autres vulnérabilités.

PROGRAMME :

1. **Exécution des LLMs :**
 - Exploration des API de LLMs, modèles open-source.
 - Techniques de prompt engineering et de structuration des sorties.
2. **Construction d'un stockage vectoriel :**
 - Ingestion et division de documents.
 - Utilisation de modèles d'embedding et bases de données vectorielles.
3. **Génération augmentée par la récupération (RAG) :**
 - Utilisation d'orchestrateurs et de récupérateurs pour améliorer la génération de contenu.
 - Techniques de mémorisation et d'évaluation des performances.
4. **Optimisation de l'inférence :**
 - Application de techniques comme Flash Attention et le cache clé-valeur.
 - Décodeur spéculatif pour accélérer la génération de texte.
5. **Déploiement des LLMs :**
 - Stratégies pour le déploiement local, serveur et en périphérie.
 - Utilisation de frameworks pour la démonstration et le déploiement à grande échelle.
6. **Sécurisation des LLMs :**
 - Techniques contre le piratage de prompt et l'implantation de backdoors.
 - Mesures défensives et surveillance en production.

Fait à Saint Etienne du Rouvray, le 10/11/2023

A handwritten signature in black ink, consisting of several overlapping loops and a long horizontal stroke extending to the right.

Cyril MARTEAUX,

Directeur de la Formation et de l'International



Partenaire stratégique
Institut Mines-Télécom



ESIGELEC
Technopôle du Madrillet - Avenue Galilée
76800 Saint-Étienne-du-Rouvray
Tél. : 02.32.91.58.58



RENSEIGNEMENTS
com@esigelec.fr

esigelec.fr