

## Module : ARCHITECTURE DES SI I

Dernière mise à jour : 07/03/2022

Code	HE	HNE	ECTS
INFASI0001	42H	30H	3

<b>Responsable Module</b>	Emna Khanfir
<b>Enseignants Intervenant</b>	Nesrine Naouar, Zeineb Ben Said, Amina Aoun, Tarek Ayari, Rahma Amri, Lassaad Saidani, Abir Lahbecha
<b>Unité pédagogique</b>	UP ASI (Architectures des Systèmes d'Information)
<b>Unité d'enseignement</b>	Architectures n-tiers
<b>Prérequis</b>	-Langage de modélisation (UML) -Algorithmique et Structures de Données -Sys. De Gestion de Bases de Données
<b>Niveaux et Options</b>	4ème année SAE, SE, TWIN, SIM, SLEAM, ERPBI, ARCTIC, DS, NIDS, INFINI, GAMIX

### Objectif du module :

A la fin de ce module l'apprenant sera capable de développer des applications à l'aide de la plateforme .Net, tout en appliquant les bonnes pratiques en programmation et les concepts des architectures n-tiers.

### Mode d'évaluation :

La moyenne de ce module est calculée comme suit :

- Contrôle continu (40%) : L'étudiant sera évalué à travers un test et des travaux à rendre à la fin de chaque chapitre.
- Examen final (60%) : L'étudiant sera évalué à travers un examen pratique.

## Acquis d'apprentissage :

à la validation de ce module l'étudiant sera capable de:

AA	Acquis d'apprentissage	Niveau d'approfondissement (*)
AA1	Identifier et découvrir une plateforme de développement et des types d'architectures logicielles.	1,3
AA2	Expliquer et préparer la couche des entités.	3,4,6
AA3	Formuler des requêtes sur une source de données via un langage de requête.	3
AA4	Expliquer, préparer et développer la couche d'accès aux données.	2,3,6
AA5	Distinguer, choisir et intégrer des patrons de conception à la solution.	4,5,6
AA6	Développer un ensemble de services pour implémenter la couche métier.	6
AA7	Expliquer et développer la couche de présentation en appliquant l'injection de dépendance.	2,3,4,6

\* : (1 : Mémoriser, 2 : Comprendre, 3 : Appliquer, 4 : Analyser, 5 : évaluer, 6 : Créer).

## Contenu détaillé

### ***Introduction à une plateforme de développement (.NET) [Chap.1]***

Décrire une plateforme de développement (.Net).

Présenter les architectures logicielles.

Découvrir une plateforme de développement (.Net).

Employer une architecture logicielle.

Situation(s) d'apprentissage	Cours intégré
Durée	3H
Rendu(s)	Installer l'environnement (Visual studio 2019).

## Principes de l'orienté objet et bases du langage C# [Chap.2 - Partie 1]

Distinguer les différents types de données (valeur et référence).

Utiliser les variables implicitement et explicitement typées.

Utiliser les structures itératives et conditionnelles.

Traiter les exceptions.

Distinguer les conversions explicites et implicites.

Utiliser des méthodes et des classes génériques.

Distinguer les types de collections (génériques et non génériques).

Appliquer les principes de l'orientée objet (encapsulation, polymorphisme, héritage).

Situation(s) d'apprentissage	Cours intégré
Durée	6h
Rendu(s)	TP1

## Concepts avancés et interrogation des sources de données en C# [Chap.2 - Partie2]

Utiliser les délégués.

Utiliser les méthodes d'extension.

Utiliser les méthodes anonymes.

Utiliser les expressions lambda.

Utiliser les méthodes d'un langage de requête (LINQ) pour interroger une source de données.

Situation(s) d'apprentissage	Cours intégré
Durée	3H
Rendu(s)	TP2

## Génération de la base de données [Chap.3 – Partie 1]

Créer la couche d'accès aux données.

Utiliser un outil de mapping entre le monde relationnel et le monde objet (ORM).

Présenter les différentes approches de l'outil de mapping.

Appliquer une approche de l'outil de mapping.

Appliquer les migrations pour générer et mettre à jour le schéma de la base de données (BD).

Représenter les conventions par défaut.

Situation(s) d'apprentissage	Cours intégré
Durée	6H
Rendu(s)	TP3

### ***Enrichir la boîte à outils d'un ORM [Chap.3 – Partie 2]***

Créer des conventions personnalisées.

Présenter et appliquer des annotations.

Présenter et créer des configurations Fluent API.

Créer un type d'entité détenu.

Créer une table porteuse de données.

Utiliser le chargement différé des données associées (Lazy Loading).

Situation(s) d'apprentissage	Cours intégré
Durée	6H
Rendu(s)	TP4-5

### ***Principes & patrons de conception [Chap.4 – Partie 1]***

Identifier et expliquer les principes S.O.L.I.D.

Expliquer les patrons de conception.

Adapter les patrons de conception dans une solution.

Situation(s) d'apprentissage	Cours intégré
Durée	3H
Rendu(s)	TP6

## Développement de la couche métier [Chap.4 – Partie 2]

Créer la couche de services génériques.

Créer la couche service.

Créer une ensemble de services permettant l'interrogation et la mise à jour d'une BD.

Situation(s) d'apprentissage	Cours intégré
Durée	3H
Rendu(s)	TP7

## Développement de la couche de présentation [Chap.5]

Créer la couche de présentation avec une plateforme de développement web (ASP.Net).

Expliquer et utiliser le patron architectural MVC.

Appliquer l'injection de dépendance.

Créer les contrôleurs et implémenter les actions.

Créer les vues et les vues partielles.

Créer des listes déroulantes (statiques et dynamiques).

Expliquer comment importer et afficher des images.

Situation(s) d'apprentissage	Cours intégré
Durée	12H
Rendu(s)	TP8

### Evaluation :

	Oral assessment	Practic al exam	Report/ Homew ork	Presen tation	TP	Project
Identifier et découvrir une plateforme de développement et des types d'architectures logicielles.		X			X	
Expliquer et préparer la couche des entités.		X			X	
Formuler des requêtes sur une source de données via un langage de requête.		X			X	

Expliquer, préparer et développer la couche d'accès aux données.		X			X	
Distinguer, choisir et intégrer des patrons de conception à la solution.		X			X	
Développer un ensemble de services pour implémenter la couche métier.		X			X	
Expliquer et développer la couche de présentation en appliquant l'injection de dépendance.		X			X	

## Références:

### Références bibliographiques :

**Ref1** : Language Integrated Query (LINQ) in C#. Microsoft Docs.  
<https://docs.microsoft.com/en-us/dotnet/csharp/linq/>

**Ref2** : Object-Oriented Programming (C#). Microsoft Docs.  
<https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/tutorials/oop>

**Ref3** : Collections (C#). Microsoft Docs.  
<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/collections>

**Ref4** : Overview of Entity Framework Core - EF Core. Microsoft Docs.  
<https://docs.microsoft.com/en-us/ef/core/>

**Ref5** : Overview of ASP.NET Core MVC. Microsoft Docs.  
<https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-6.0>