



National University of Engineering (UNI)
School of Cybersecurity
Syllabus 2024-II

1. COURSE

CS1D2. Discrete Structures II (Mandatory)

2. GENERAL INFORMATION

2.1 Course	:	CS1D2. Discrete Structures II
2.2 Semester	:	2 nd Semester.
2.3 Credits	:	4
2.4 Horas	:	2 HT; 4 HP;
2.5 Duration of the period	:	16 weeks
2.6 Type of course	:	Mandatory
2.7 Learning modality	:	Face to face
2.8 Prerequisites	:	CS1D1. Discrete Structures I. (1 st Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

In order to understand the advanced computational techniques, the students must have a strong knowledge of the Various discrete structures, structures that will be implemented and used in the laboratory in the programming language..

5. GOALS

- That the student is able to model computer science problems using graphs and trees related to data structures.
- That the student applies efficient travel strategies to be able to search data in an optimal way.
- That the student uses the various counting techniques to solve computational problems.

6. COMPETENCES

1) ()

6) Apply security principles and practices to maintain operations in the presence of risks and threats.()

7. TOPICS

Unit 1: Digital Logic and Data Representation (10 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> • Reticles: Types and properties. • Boolean algebras. • Boolean Functions and Expressions. • Representation of Boolean Functions: Normal Disjunctive and Conjunctive Form. • Logical gates. • Circuit Minimization. 	<ul style="list-style-type: none"> • Explain the importance of Boolean algebra as a unification of set theory and propositional logic [Evaluar]. • Explain the algebraic structures of reticulum and its types [Evaluar]. • Explain the relationship between the reticulum and the ordinate set and the wise use to show that a set is a reticulum [Evaluar]. • Explain the properties that satisfies a Boolean algebra [Evaluar]. • Demonstrate if a terna formed by a set and two internal operations is or not Boolean algebra [Evaluar]. • Find the canonical forms of a Boolean function [Evaluar]. • Represent a Boolean function as a Boolean circuit using logic gates [Evaluar]. • Minimize a Boolean function. [Evaluar].
Readings : [Rosen2007], [Grimaldi03]	

Unit 2: Fundamentos de conteo (40 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> • Técnicas de Conteo: <ul style="list-style-type: none"> – Conteo y cardinalidad de un conjunto – Regla de la suma y producto – Principio de inclusión-exclusión – Progresión geométrica y aritmética • Principio de las casillas. • Permutaciones y combinaciones: <ul style="list-style-type: none"> – Definiciones básicas – Identidad de Pascal – Teorema del binomio • Resolviendo relaciones de recurrencia: <ul style="list-style-type: none"> – Un ejemplo de una relación de recurrencia simple, como los números de Fibonacci – Otros ejemplos, mostrando una variedad de soluciones • Aritmetica modular basica 	<ul style="list-style-type: none"> • Aplicar argumentos de conteo, incluyendo las reglas del producto y de la suma, principio de inclusión-exclusión y progresiones aritméticas/geométricas [Familiarizarse] • Aplicar el principio de las casillas en el contexto de una demostración formal [Familiarizarse] • Calcular permutaciones y combinaciones en un conjunto, e interpreta su significado en el contexto de una aplicación en particular [Familiarizarse] • Mapear aplicaciones del mundo real a formalismos de conteo adecuados, como el determinar el número de formas de acomodar a un conjunto de personas alrededor de una mesa, sujeto a restricciones en la disposición de los asientos, o en el número de maneras de determinar ciertas manos en juegos de cartas (ejm. una casa llena) [Familiarizarse] • Resolver una variedad de relaciones de recurrencia básicas [Familiarizarse] • Analizar un problema para determinar las relaciones de recurrencia implícitas [Familiarizarse] • Realizar cálculos que involucran aritmética modular [Familiarizarse]
Readings : [Grimaldi97]	

Unit 3: Árboles y Grafos (40 hours)	
Competences Expected:	
Topics	Learning Outcomes
<ul style="list-style-type: none"> • Árboles. <ul style="list-style-type: none"> – Propiedades – Estrategias de recorrido • Grafos no dirigidos • Grafos dirigidos • Grafos ponderados • Arboles de expansion/bosques. • Isomorfismo en grafos. 	<ul style="list-style-type: none"> • Ilustrar mediante ejemplos la terminología básica de teoría de grafos, y de alguna de las propiedades y casos especiales de cada tipo de grafos/árboles [Familiarizarse] • Demostrar diversos métodos de recorrer árboles y grafos, incluyendo recorridos pre, post e inorden de árboles [Familiarizarse] • Modelar una variedad de problemas del mundo real en ciencia de la computación usando formas adecuadas de grafos y árboles, como son la representación de una topología de red o la organización jerárquica de un sistema de archivos [Familiarizarse] • Demostrar como los conceptos de grafos y árboles aparecen en estructuras de datos, algoritmos, técnicas de prueba (inducción estructurada), y conteos [Familiarizarse] • Explicar como construir un árbol de expansión de un grafo [Familiarizarse] • Determinar si dos grafos son isomorfos [Familiarizarse]
Readings : [Johnsonbaugh99]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

***** EVALUATION MISSING *****

10. BASIC BIBLIOGRAPHY