National University of Engineering (UNI)
School of Computer Science
Sillabus 2023-I

1. COURSE

CS1D2. Discrete Structures II (Mandatory)

2. GENERAL INFORMATION

2.1 Course	:	CS1D2. Discrete Structures II
2.2 Semester	:	2^{do} Semestre.
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	:	Blended
2.8 Prerrequisites	:	CS1D1. Discrete Structures I. (1^{st} Sem) 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) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions. (**Familiarity**)

6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (**Familiarity**)

7. TOPICS

Unit 1: Digital Logic and Data Representation (10)	
Competences Expected:	
Topics	**Learning Outcomes**
• Reticles: Types and properties. • Boolean algebras. • Boolean Functions and Expressions. • Representation of Boolean Functions: Normal Disjunctive and Conjunctive Form. • Logical gates. • Circuit Minimization.	• Explain the importance of Boolean algebra as a unification of set theory and propositional logic [Assessment]. • Explain the algebraic structures of reticulum and its types [Assessment]. • Explain the relationship between the reticulum and the ordinate set and the wise use to show that a set is a reticulum [Assessment]. • Explain the properties that satisfies a Boolean algebra [Assessment]. • Demonstrate if a terna formed by a set and two internal operations is or not Boolean algebra [Assessment]. • Find the canonical forms of a Boolean function [Assessment]. • Represent a Boolean function as a Boolean circuit using logic gates [Assessment]. • Minimize a Boolean function. [Assessment].
Readings : [**Rosen2007**], [Gri03]	

<table>
<tr><td colspan="2">Unit 2: Basics of Counting (40)</td></tr>
<tr><td colspan="2">Competences Expected:</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

Counting arguments

Set cardinality and counting
Sum and product rule
Inclusion-exclusion principle
Arithmetic and geometric progressions

The pigeonhole principle
Permutations and combinations

Basic definitions
Pascal's identity
The binomial theorem

Solving recurrence relations

An example of a simple recurrence relation, such as Fibonacci numbers
Other examples, showing a variety of solutions

Basic modular arithmetic

</td><td>

Apply counting arguments, including sum and product rules, inclusion-exclusion principle and arithmetic/geometric progressions [Familiarity]
Apply the pigeonhole principle in the context of a formal proof [Familiarity]
Compute permutations and combinations of a set, and interpret the meaning in the context of the particular application [Familiarity]
Map real-world applications to appropriate counting formalisms, such as determining the number of ways to arrange people around a table, subject to constraints on the seating arrangement, or the number of ways to determine certain hands in cards (eg, a full house) [Familiarity]
Solve a variety of basic recurrence relations [Familiarity]
Analyze a problem to determine underlying recurrence relations [Familiarity]
Perform computations involving modular arithmetic [Familiarity]

</td></tr>
<tr><td colspan="2">Readings : [Gri97]</td></tr>
</table>

<table>
<tr><td colspan="2">Unit 3: Graphs and Trees (40)</td></tr>
<tr><td colspan="2">Competences Expected:</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

Trees

Properties
Traversal strategies

Undirected graphs
Directed graphs
Weighted graphs
Spanning trees/forests
Graph isomorphism

</td><td>

Illustrate by example the basic terminology of graph theory, and some of the properties and special cases of each type of graph/tree [Familiarity]
Demonstrate different traversal methods for trees and graphs, including pre, post, and in-order traversal of trees [Familiarity]
Model a variety of real-world problems in computer science using appropriate forms of graphs and trees, such as representing a network topology or the organization of a hierarchical file system [Familiarity]
Show how concepts from graphs and trees appear in data structures, algorithms, proof techniques (structural induction), and counting [Familiarity]
Explain how to construct a spanning tree of a graph [Familiarity]
Determine if two graphs are isomorphic [Familiarity]

</td></tr>
<tr><td colspan="2">Readings : [Joh99]</td></tr>
</table>

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

[Gri03] R. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction*. 5 ed. Pearson, 2003.

[Gri97] R. Grimaldi. *Matemáticas Discretas y Combinatoria*. Addison Wesley Iberoamericana, 1997.

[Joh99] Richard Johnsonbaugh. *Matemáticas Discretas*. Prentice Hall, México, 1999.