



Universidad de Ingeniería y Tecnología
Escuela Profesional de
Ciencia de la Computación
Silabo del curso
Periodo Académico 2019-I

1. **Código del curso y nombre:** CS2101. Teoría de la Computación (Obligatorio)

2. **Créditos:** 4

3. **Horas de Teoría y Laboratorio:** 2 HT; 4 HL; (Semanal)

4. **Profesor(es) del curso, email y horario de atención**

Titular

- Juan Gutierrez Alva <jgutierrez@utec.edu.pe>
– Doctor en Ciencia de la Computación, Universidad de São Paulo, Brasil, 2018.

Atención previa coordinación con el profesor

5. **Bibliografía básica**

[Bro93] J. Glenn Brookshear. *Teoría de la Computación*. Addison Wesley Iberoamericana, 1993.

[HU93] John E. Hopcroft and Jeffrey D. Ullman. *Introducción a la Teoría de Autómatas, Lenguajes y Computación*. CECSA, 1993.

[Kel95] Dean Kelley. *Teoría de Autómatas y Lenguajes Formales*. Prentice Hall, 1995.

[Kol97] Ross Kolman Busby. *Estructuras de Matemáticas Discretas para la Computación*. Prentice Hall, 1997.

6. **Información del curso**

- (a) **Breve descripción del curso** Este curso hace énfasis en los lenguajes formales, modelos de computación y computabilidad, además de incluir fundamentos de la complejidad computacional y de los problemas NP completos.
- (b) **Prerrequisitos:** CS1D02. Estructuras Discretas II. (2^{do} Sem)
- (c) **Tipo de Curso:** Obligatorio
- (d) **Modalidad:** Presencial

7. **Objetivos del curso.**

Competencias

- a) Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (**Evaluar**)
- b) Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución. (**Evaluar**)
- j) Aplicar la base matemática, principios de algoritmos y la teoría de la Ciencia de la Computación en el modelamiento y diseño de sistemas computacionales de tal manera que demuestre comprensión de los puntos de equilibrio involucrados en la opción escogida. (**Evaluar**)

Objetivos de Aprendizaje

- Que el alumno aprenda los conceptos fundamentales de la teoría de lenguajes formales.



8. **Tópicos del curso**

1. Computabilidad y complejidad básica de autómatas
2. Complejidad Computacional Avanzada
3. Teoría y Computabilidad Avanzada de Autómatas

9. Metodología y sistema de evaluación

Sesiones Teóricas:

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

Exposiciones individuales o grupales:

Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

Lecturas:

A lo largo del curso se proporcionan diferentes lecturas, las cuales son evaluadas. El promedio de las notas de las lecturas es considerado como la nota de una práctica calificada. El uso del campus virtual UTEC Online permite a cada estudiante acceder a la información del curso, e interactuar fuera de aula con el profesor y con los otros estudiantes.

Sistema de Evaluación:

La nota final NF se obtiene a través de:

$$NF = 0.2 * E_1 + 0.2 * E_2 + \\ 0.2 * E_3 + 0.2 * E_4 + \\ 0.2 * P$$

Donde:

E: Examen (4), Todos los textos requieren impresión

P: Proyecto (1)

Para aprobar el curso hay que obtener 11 o más en la nota final NF .

10. Contenido



Unidad 1: Computabilidad y complejidad básica de autómatas (20)	
Competences esperadas: C9	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Discute el concepto de máquina de estado finito [Evaluar] • Diseña una máquina de estado finito determinista para aceptar un determinado lenguaje [Evaluar] • Genere una expresión regular para representar un lenguaje específico [Evaluar] • Explique porque el problema de la parada no tiene solución algorítmica [Evaluar] • Diseña una gramática libre de contexto para representar un lenguaje especificado [Evaluar] • Define las clases P y NP [Evaluar] • Explique el significado de NP-Complejidad [Evaluar] • Explica la tesis de Church-Turing y su importancia [Familiarizarse] • Explica el teorema de Rice y su importancia [Familiarizarse] • Da ejemplos de funciones no computables [Familiarizarse] • Demuestra que un problema es no computable al reducir un problema clásico no computable en base a él [Familiarizarse] 	<ul style="list-style-type: none"> • Máquinas de estado finito. • Expresiones regulares. • Problema de la parada. • Gramáticas libres de contexto. • Introducción a las clases P y NP y al problema P vs. NP. • Introducción y ejemplos de problemas NP- Completos y a clases NP-Complejos. • Máquinas de Turing, o un modelo formal equivalente de computación universal. • Máquinas de Turing no determinísticas. • Jerarquía de Chomsky. • La tesis de Church-Turing. • Computabilidad. • Teorema de Rice. • Ejemplos de funciones no computables. • Implicaciones de la no-computabilidad.
Lecturas : [Kol97], [Kel95]	

Unidad 2: Complejidad Computacional Avanzada (20)	
Competences esperadas: C8,C9	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Define las clases P y NP (También aparece en AL / Automata Básico, Computabilidad y Complejidad) [Evaluar] • Define la clase P-Space y su relación con la clase EXP [Evaluar] • Explique el significado de NP-Completo (También aparece en AL / Automata Básico, Computabilidad y Complejidad) [Evaluar] • Muestre ejemplos de problemas clásicos en NP - Completo [Evaluar] • Pruebe que un problema es NP- Completo reduciendo un problema conocido como NP-Completo [Evaluar] 	<ul style="list-style-type: none"> • Revisión de las clases P y NP; introducir espacio P y EXP. • Jerarquía polinomial. • NP completitud (Teorema de Cook). • Problemas NP completos clásicos. • Técnicas de reducción.
Lecturas : [Kel95], [HU93]	



Unidad 3: Teoría y Computabilidad Avanzada de Autómatas (20)	
Competences esperadas: C8	
Objetivos de Aprendizaje	Tópicos
<ul style="list-style-type: none"> • Determina la ubicación de un lenguaje en la jerarquía de Chomsky (regular, libre de contexto, enumerable recursivamente) [Evaluar] • Convierte entre notaciones igualmente poderosas para un lenguaje, incluyendo entre estas AFDs, AFNDs, expresiones regulares, y entre AP y GLCs [Evaluar] 	<ul style="list-style-type: none"> • Conjuntos y Lenguajes: <ul style="list-style-type: none"> – Lenguajes Regulares. – Revisión de autómatas finitos determinísticos (Deterministic Finite Automata DFAs) – Autómata finito no determinístico (Nondeterministic Finite Automata NFAs) – Equivalencia de DFAs y NFAs. – Revisión de expresiones regulares; su equivalencia con autómatas finitos. – Propiedades de cierre. – Probando no-regularidad de lenguajes, a través del lema de bombeo (Pumping Lemma) o medios alternativos. • Lenguajes libres de contexto: <ul style="list-style-type: none"> – Autómatas de pila (Push-down automata (PDAs) – Relación entre PDA y gramáticas libres de contexto. – Propiedades de los lenguajes libres de contexto.
Lecturas : [HU93], [Bro93]	

