UNIVERSIDAD NACIONAL
DE
SANTIAGO DEL ESTERO

FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍAS

PLANIFICACIÓN ANUAL 2025

ASIGNATURA: Teoría de la Computación

LICENCIATURA EN SISTEMAS DE INFORMACIÓN
Plan de Estudios: 2011 – Innovación Curricular 2022

Equipo cátedra:

Profesor Titular: Msc. Ing. Margarita María Álvarez

JTP: Dra. Paola Budán

1. IDENTIFICACIÓN

1.1. NOMBRE DE LA ASIGNATURA: TEORÍA DE LA COMPUTACIÓN
1.2. CARRERA: LICENCIATURA EN SISTEMAS DE INFORMACIÓN
1.3. PLAN DE ESTUDIOS: 2011 Innovación Curricular 2022
1.4. AÑO ACADÉMICO: 2025
1.5. CARÁCTER: Obligatorio
1.6. UBICACIÓN DE LA ASIGNATURA EN EL PLAN DE ESTUDIOS:

1.6.1. MÓDULO: 5° - **AÑO:** 3°

1.6.2.TRAYECTO AL QUE PERTENECE LA ASIGNATURA

Tabla 1: Carga horaria

TRAYECTO	CARGA HORARIA PRESENCIAL
Ciencias Básicas Generales y Específicas	75 hs
Algoritmos y Lenguajes	
Arquitectura, Sistemas Operativos y Redes	
Ingeniería de Software, Bases de Datos y Sistemas de Información	
Aspectos Sociales y Profesionales	
Otros contenidos	
CARGA HORARIA TOTAL DE LA ACTIVIDAD CURRICULAR	75 hs.

1.6.3-CORRELATIVAS

1.6.3.1 Anteriores:

- INGLÉS I (REGULAR)
- ESTRUCTURAS DE DATOS Y PROGRAMACIÓN II (REGULAR)
- LÓGICA II (REGULAR)
- ÁLGEBRA I (Anual) (APROBADA)
- ANÁLISIS I (Anual) (APROBADA)

1.6.3.2. Posteriores: LENGUAJES DE PROGRAMACIÓN Y COMPILADORES (APROBADA)

1.7- CARGA HORARIA:

1.7.1. **Carga horaria semanal total**: 5 hs
1.7.2. **Carga horaria semanal destinada a la formación práctica**: 3 hs.
1.7.3 **Carga horaria total dedicada a las distintas actividades de formación práctica:** 45 hs.

1.8. ÁMBITOS DONDE SE DESARROLLAN LAS ACTIVIDADES DE FORMACIÓN PRÁCTICA: aulas.

1.9. CANTIDAD DE COMISIONES EN LAS QUE SE DICTA LA ASIGNATURA: 1

2. PRESENTACIÓN

2.1. UBICACIÓN DE LA COMO TRAMO DE CONOCIMIENTO DE UNA DISCIPLINA

Esta asignatura se ubica en el trayecto de Ciencias básicas generales y específicas. Se aborda el núcleo teórico fundamental de las Ciencias de la Computación: la Teoría de la Computación. Bajo este nombre se recogen una serie de materias que constituyen hoy en día los fundamentos teóricos de la Informática, a saber: la Teoría de la Computabilidad y Complejidad Algorítmica, la Teoría de Autómatas y la Teoría de los Lenguajes Formales.

La Teoría de la Computabilidad estudia los problemas de decisión que pueden ser resueltos por un algoritmo (o equivalentemente por una máquina de Turing) y explora las limitaciones de las computadoras al establecer qué tipos de problemas pueden ser resueltos por una máquina.

Con el estudio de la Teoría de la Complejidad Algorítmica, el alumno dispone de un marco de referencia para clasificar problemas, saber qué modelo de cálculo requieren y lograr diseñar algoritmos eficientes, según el tipo de problemas.

También, se ofrece una introducción a la Teoría de Lenguajes Formales y Gramáticas, cuyo núcleo consiste en la jerarquía de lenguajes definida por Chomsky y sus modelos de representación.

Paralelamente a la jerarquía de lenguajes existe otra equivalencia de máquinas abstractas, de tal forma que a cada una de las clases de lenguajes definidas en la jerarquía de Chomsky, a partir de restricciones impuestas a las gramáticas, les corresponde un tipo de máquina abstracta, que no es otra cosa que un método reconocedor de lenguajes.

El valor formativo de todos los conceptos desarrollados en la asignatura es esencial en dos aspectos: para sustentar una posterior profundización de la Teoría de la Computación y el posterior estudio de Compiladores e Intérpretes y, además, para proveer una base sólida para el cabal ejercicio profesional.

2.2. CONOCIMIENTOS Y HABILIDADES PREVIAS QUE PERMITEN ENCARAR EL APRENDIZAJE DE LA ASIGNATURA

Los prerrequisitos para el abordaje de esta asignatura tienen que ver con los conocimientos y habilidades adquiridos con la Teoría de Conjuntos, el Cálculo Proposicional y la Programación.

2.3. ASPECTOS DEL PERFIL PROFESIONAL DEL EGRESADO A LOS QUE CONTRIBUYE LA ASIGNATURA

El Licenciado en Sistemas de Información es un profesional que:

- Posee:
  - Los conocimientos básicos (lógico-matemáticos y computacionales) para una adecuada fundamentación teórica de su quehacer profesional específico.

Está capacitado para:

- Identificar, formular y resolver problemas de Informática.
- Gestar, diseñar y desarrollar proyectos de Informática.
- Utilizar técnicas y herramientas de aplicación en la Informática.
- Comunicarse de manera efectiva en el ámbito profesional.
- Tiene una actitud flexible para integrar equipos interdisciplinarios en el desarrollo y administración de proyectos de Sistemas Informáticos y de Sistemas de Información.

2.4- INTEGRACIÓN HORIZONTAL Y VERTICAL CON OTRAS ASIGNATURAS.

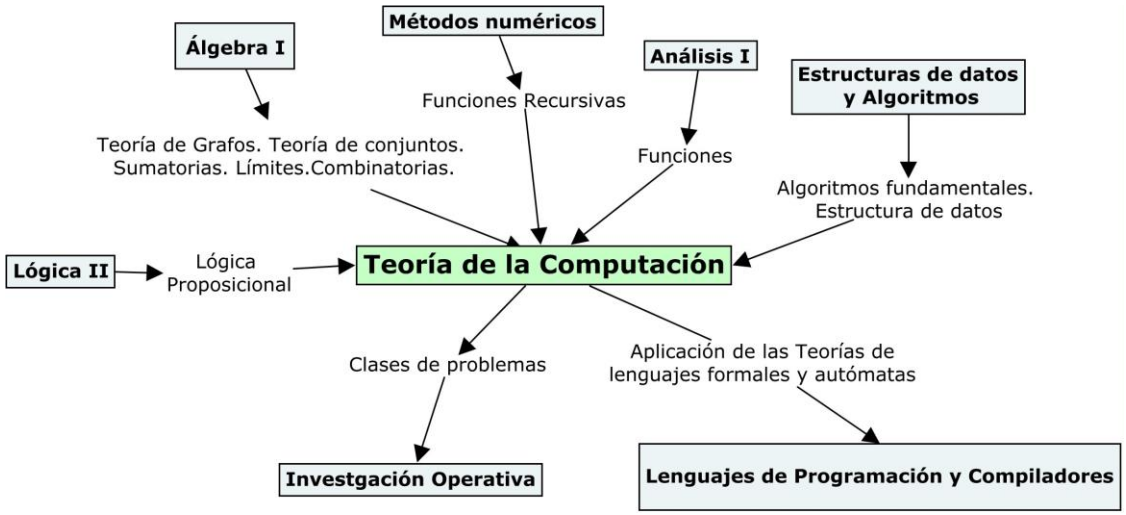

Figura 1. **Integración horizontal y vertical con otras asignaturas**

3. OBJETIVOS

1. Formalizar problemas de decisión y evaluar la Computabilidad.
2. Determinar la eficiencia de Algoritmos y su Complejidad computacional usando medidas de complejidad.
3. Reconocer y definir gramáticas y lenguajes de distinto tipo basados en la Teoría de lenguajes y gramáticas.
4. Reconocer y Definir autómatas finitos, de pila y máquinas de Turing (como reconocedora de lenguajes y como procedimiento) utilizando la Teoría de Autómatas.
5. Diseñar los dispositivos básicos para generar un lenguaje de programación sencillo (gramáticas, expresiones regulares de base para la construcción de compiladores / intérpretes).

4. SELECCIÓN Y ORGANIZACIÓN DE CONTENIDOS

4.1 CONTENIDOS MÍNIMOS ESTABLECIDOS EN EL PLAN DE ESTUDIOS

Evaluación de Computabilidad. Conceptos Básicos de Teoría de Computabilidad. Problemas computables y no computables. Problema de la detención. Complejidad computacional. Análisis de Algoritmos: Análisis asintótico, comportamiento en el mejor caso, caso promedio y peor caso. Notación O(). Problemas tratables e intratables. Clases de complejidad. Fundamentos de Autómatas y Gramáticas: Lenguajes formales, Gramáticas. Jerarquía de Chomsky. Propiedades de las gramáticas. Expresiones regulares. Autómatas finitos, autómatas de pila y máquinas de Turing. Relación entre gramáticas y autómatas.

4.2. PROGRAMA SINTÉTICO SOBRE LA BASE DE LOS CONTENIDOS MÍNIMOS

1. **Teoría de la Computabilidad**. Conceptos Básicos de Teoría de Computabilidad. Formalización, representación y solución de problemas. Problemas computables y no computables. Función computable.
2. **Teoría de la Complejidad Computacional**. Análisis de Complejidad de Algoritmos. Análisis de Algoritmos: Análisis asintótico, comportamiento en el mejor caso, caso promedio y peor caso. Notación O(). Balance entre tiempo y espacio en los algoritmos. Clasificación de problemas: Problema de la detención. Problemas tratables e intratables.
3. **Teoría de los Lenguajes Formales**. Definición y conceptos. Gramáticas de estructura de frase. Jerarquía de Chomsky. Propiedades de las gramáticas. Expresiones regulares
4. **Teoría de Autómatas**. Autómatas finitos, Autómata a pila. Máquina de Turing. Interpretaciones: como reconocedora de lenguajes y como procedimiento. Relación entre gramáticas y autómatas.

4.3 ARTICULACIÓN TEMÁTICA DE LA ASIGNATURA

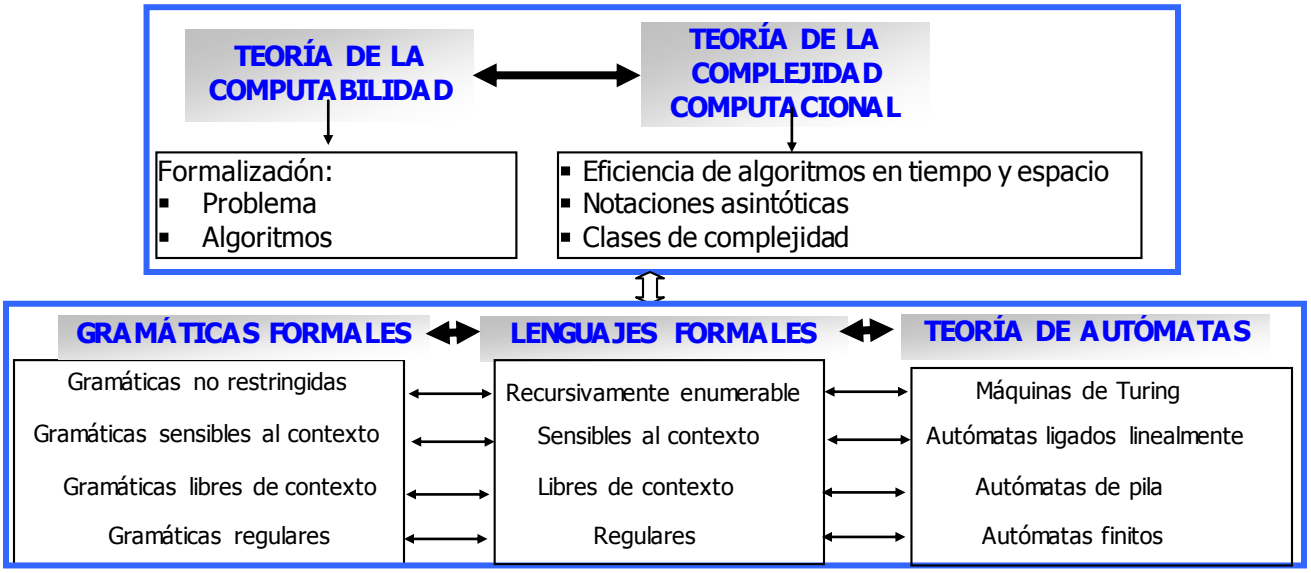

Figura 2: **Articulación temática de la asignatura**

4.4. PROGRAMA ANALÍTICO

1. TEORÍA DE LA COMPUTABILIDAD

i. Concepto de problema. Formalización, representación y solución de problemas. Procedimientos y algoritmos. Problemas computables y no computables.

ii. ***Conceptos básicos de la teoría:*** computabilidad, enumerabilidad, decidibilidad y generabilidad. Función computable. Problema de la detención.

2. TEORÍA DE LA COMPLEJIDAD COMPUTACIONAL

i. **COMPLEJIDAD Y EFICIENCIA DE ALGORITMOS**: Teoría de la Complejidad Computacional. Análisis de algoritmos: Análisis asintótico, comportamiento en el mejor caso, caso promedio y peor caso. Orden de un algoritmo. Notación 0(). Principios y consideraciones para la determinación del orden. Balance entre tiempo y espacio en los algoritmos. Problemas tratables e intratables.

ii. **CLASIFICACIÓN DE PROBLEMAS:** Problemas demostrablemente insolubles, demostrablemente difíciles. Clase P, Clase NP, NP completa y CO-NP.

3. TEORÍA DE LENGUAJES FORMALES

i. **CONCEPTOS BÁSICOS**: Símbolo, alfabeto, palabra. Operaciones con hileras. Operaciones con lenguajes: unión, intersección, concatenación. Propiedades de las operaciones. Isomorfismo

ii. **GRAMÁTICAS FORMALES:** definición, tipos de gramáticas. Jerarquía de Chomsky. Gramáticas no restringidas, sensibles al contexto, libres de contexto y regulares. Lenguajes generados por cada tipo de gramáticas.

iii. **CARACTERÍSTICAS DE LAS GRAMÁTICAS**: Gramáticas libres de contexto: árbol de derivación, derivaciones a izquierda y a derecha, recursividad a izquierda directa e indirecta, factorización y ambigüedad. Gramáticas propias.

iv. **GRAMÁTICAS REGULARES**: Expresiones regulares. Propiedades y equivalencias.

4. TEORÍA DE AUTÓMATAS

i. **AUTÓMATAS FINITOS:** Definición y representación gráfica. El autómata finito como reconocedor de lenguajes. Autómata finito determinista y no determinista. Equivalencia. Minimización de autómatas finitos deterministas.

ii. **AUTÓMATA DE PILA:** Definición formal. Autómata de pila como reconocedor de un lenguaje. Autómata a pila determinístico y no determinístico.

iii. **MÁQUINAS DE TURING**: Definición formal. Representación. Interpretaciones de las computaciones. Configuración de una máquina de Turing. Máquina de Turing multicinta. Máquina universal de Turing. Codificación de una máquina de Turing.

4.5. CRONOGRAMA PARA EL DESARROLLO DE LAS UNIDADES TEMÁTICAS

Tabla 2: Cronograma para el desarrollo teórico de las unidades temáticas

UNIDAD	CARGA HORARIA	CRONOGRAMA DE DICTADO
UNIDAD 1: TEORÍA DE LA COMPUTABILIDAD	**2hs.**	27 / 03
UNIDAD 2: TEORÍA DE LA COMPLEJIDAD COMPUTACIONAL	**8 hs.**	03 / 04
		10/ 04
		17 / 04
		24/ 04
UNIDAD 3: TEORÍA DE LENGUAJES FORMALES	**8 hs.**	8 / 05
		15 / 05
		22 / 05
		29 / 05
		5 / 06

UNIDAD	CARGA HORARIA	CRONOGRAMA DE DICTADO
UNIDAD 4: TEORÍA DE AUTÓMATAS	**8 hs.**	12 / 06
		19 / 06
		26 /06
TOTAL	**26 hs.**	

5. FORMACIÓN PRÁCTICA

5.1. Descripción de las actividades de formación práctica

Durante el cursado los alumnos realizan trabajos prácticos y una actividad de resolución de problemas del mundo real, que les permiten interpretar e intervenir de manera efectiva para resolver problemas relacionados con la aplicación de la Teoría de la Computabilidad.

Trabajo Práctico N° 1: *Teoría de la Computabilidad y de la Complejidad*

Actividades. Se plantean los siguientes tipos de ejercicios:

1) Identificar problemas de decisión.
2) Proponer ejemplos de problemas de decisión.
3) Formalizar problemas de decisión mediante una cuádrupla en la que la condición de viabilidad tiene que estar expresada con precisión matemática y/o lógica.
4) Identificar problemas tratables e intratables mediante la eficiencia de algoritmos. Para ello, debe determinar la eficiencia de un algoritmo y obtener el tiempo de ejecución del peor caso.
5) Realizar cálculos del orden de cada algoritmo para aproximar una eficiencia polinomial.
6) Desarrollar un programa sobre un algoritmo de Markov que resuelva un problema sencillo.
7) Investigar en sitios web actualizados sobre la clasificación de problemas y extraer ejemplos para las clases P, NP y NP-completa.
8) *Trabajo de investigación*, el cual se comunica por escrito, para complementar el estudio de la eficiencia de los algoritmos planteado en el Trabajo Práctico N° 1, mediante la interpretación de papers actuales.

Trabajo Práctico N° 2: *Teoría de Lenguajes Formales y gramáticas*

Actividades. Se plantean los siguientes tipos de ejercicios:

1) Identificar lenguajes formales e hileras pertenecientes al lenguaje.
2) Obtener hileras a partir de las operaciones de: concatenación, potencia e inversa.
3) Obtener lenguajes formales a partir de las operaciones de: unión, intersección, diferencia, complemento, concatenación, potencia, inversa y cerradura de Kleene.
4) Identificar de qué tipo son los lenguajes planteados (lenguajes recursivamente enumerables, sensibles al contexto, libres de contexto y regulares) y los tipos de gramáticas que los generan.
5) Resolver el problema de generación de lenguajes formales mediante gramáticas**.**
6) Definir expresiones regulares para cada lenguaje enunciado en la actividad.
7) Generar hileras para cada lenguaje y comprobarlas utilizando JFLAP.
8) Obtener una gramática equivalente a partir de una expresión regular.
9) Aplicar los algoritmos para simplificar gramáticas.
10) Desarrollar programas de los algoritmos de simplificación.

Trabajo Práctico N° 3: *Teoría de Autómatas*

Actividades. Se plantean los siguientes tipos de ejercicios:

1) Identificar los tipos de autómatas (Máquina de Turing, Pila y Finito) y su relación con los tipos de lenguajes que reconocen cada autómata.
2) Resolver el problema de reconocimiento de lenguajes mediante autómatas (Máquina de Turing, Pila y Finito).

3) Aplicar los algoritmos de estados mínimos y de transformación de autómata finito no determinístico a determinístico.
4) Resolver problemas Turing computables tanto con Máquinas de Turing unicinta como multicintas.
5) Comprobar el funcionamiento de los dispositivos abstractos utilizando JFLAP.
6) Programar los algoritmos de conversión de un autómata finito no determinístico a determinístico y de estados mínimos.

ACTIVIDADES DE RESOLUCIÓN DE PROBLEMAS DEL MUNDO REAL: *Diseño de dispositivos para generar un lenguaje de programación*

CONTENIDOS: los contenidos temáticos necesarios para abordar la actividad de resolución de problemas del mundo real son los descriptos en la unidad 3. A continuación se los detalla sintéticamente, los más importantes: (1) Lenguajes Libre de Contexto y regulares; (2) Gramáticas Libre de Contexto y regulares.

OBJETIVOS:

- Introducir al alumno en los problemas concretos que se enfrentan durante el diseño de lenguajes de programación.
- Reforzar, desde el punto de vista práctico, los conocimientos adquiridos sobre lenguajes y gramáticas Libre de Contexto y regulares.
- Desarrollar la capacidad para diseñar los dispositivos básicos para generar instrucciones de lenguajes de programación sencillos.

ACTIVIDADES:

a) Identificar y definir el lenguaje formal. Para ello:
- Especificar el vocabulario.
- Identificar los componentes léxicos tales como: identificadores, palabras claves, constantes, etc.
- Realizar las expresiones regulares que reconozcan los componentes léxicos.

b) Resolver el problema de la generación del lenguaje. Para ello:
- Elaborar la gramática libre de contextos.
- Desarrollar la prueba con JFLAG.

c) Elaborar informe final técnico. La información debe estar debidamente organizada de forma tal de facilitar la comprensión a los destinatarios, con la utilización de los recursos gráficos y los medios necesarios para comunicar de forma efectiva y, con un adecuado uso de las normas gramaticales y con bibliografía actualizada y debidamente referenciada.

d) Exponer en grupo el trabajo realizado en el taller. Para la presentación: organice la información para facilitar la comprensión a los destinatarios, utilice los recursos gráficos y emplee adecuadamente técnicas de comunicación oral.

RECURSOS TECNOLÓGICOS: JAVA: Lenguaje de programación y JFLAG

EVALUACIÓN: El puntaje máximo es de **100 puntos**. Si bien el trabajo puede ser realizado y presentado en forma grupal, la calificación final es individual. El puntaje mínimo para **aprobar** es de **cincuenta (50) puntos**.

CRITERIOS DE EVALUACIÓN

a) Completitud de las operaciones exigidas.
b) Correctitud en las expresiones regulares y la gramática libre de contexto.
c) Cantidad y pertinencia de las pruebas realizadas.
d) Uso adecuado de JFLAP para demostrar los árboles sintácticos.

EVALUACIÓN RECUPERATORIA: en caso de no obtener el puntaje requerido en cada uno de los apartados precedentes se otorgará una sola alternativa recuperatoria, sobre temas y modalidad a determinar por los docentes de la cátedra.

CRONOGRAMA: La presentación del trabajo se realiza en el mes de junio, al finalizar el cursado.

5.2.-CRITERIOS DE EVALUACIÓN GENERALES

Los siguientes criterios de evaluación se aplican al Taller: *Diseño de dispositivos para generar un lenguaje de programación* y a los trabajos prácticos en lo que respecta a la comunicación escrita y oral y al desempeño en equipos de trabajo:

1. Entrega del informe en tiempo y forma: se utilizarán lo siguientes criterios de evaluación:
   a. Organización coherente de la información y pertinencia de los temas incluidos.
   b. Correcta redacción, utilización de lenguaje técnico y uso adecuado del estilo de escritura que facilite la comprensión.
   c. Inclusión de fuentes de información o referencias acorde a las normas APA.
   d. Presentación coherente de los resultados y conclusiones.
2. Exposición de trabajos - Coloquio individual: Se evaluará en este caso la correcta expresión oral y la solvencia para responder las preguntas, utilizando los siguientes criterios:
   a. Respuesta de forma clara y precisa a las preguntas o comentarios.
   b. Utilización correcta de técnicas de comunicación oral y de vocabulario técnico.
   c. Correcto uso de recursos gráficos y con técnicas de comunicación oral y una adecuada organización de la información.
3. Trabajo en grupo: se evaluarán de acuerdo con los siguientes criterios de evaluación:
   a. Correctitud en la planificación y asignación equitativa de tareas.
   b. Cumplimiento de los plazos establecidos.
   c. Adecuado nivel de involucramiento de cada miembro en las tareas asignadas.
   d. Adecuada comunicación entre los miembros del equipo.
   e. Realización de una autoevaluación del desempeño de cada miembro del equipo.

5.3.-Formación en Ejes Transversales

Eje	Actividades	Resultados de Aprendizaje	Grado de Profundidad en el tratamiento
Identificación, formulación y resolución de problemas de informática	**Trabajo Práctico N° 1**: *Teoría de la Computabilidad y de la Complejidad* • Identificar problemas de decisión. • Proponer ejemplos de problemas de decisión. • Formalizar problemas de decisión mediante una cuádrupla en la que la condición de viabilidad tiene que estar expresada con precisión matemática y/o lógica. • Identificar problemas tratables e intratables mediante la determinación de la eficiencia del algoritmo.	• Identifica problemas de decisión para consolidar conceptos abstractos de clasificación de problemas. • Formaliza problemas de decisión para afianzar sus habilidades de expresión matemática –lógica. • Identifica problemas tratables e intratables mediante la eficiencia de algoritmos.	**Alto**
	Trabajo Práctico N° 2: *Teoría de Lenguajes Formales y Gramáticas* • Identificar lenguajes formales e hileras pertenecientes al lenguaje. • Identificar de qué tipo son los lenguajes planteados (lenguajes recursivamente enumerables, sensibles al contexto, libres de contexto y regulares) y los tipos de gramáticas que los generan. • Obtener lenguajes formales a partir de las operaciones de: unión, intersección, diferencia, complemento, concatenación, potencia, inversa y cerradura de Kleene. • Obtener hileras a partir de las operaciones de: concatenación, potencia e inversa. • Resolver el problema de generación de lenguajes formales mediante gramáticas**.**	• Identifica lenguajes formales para poder diferenciar sus características según la expresividad de cada uno. • Opera con hileras para obtener nuevos hileras y verificar propiedades matemáticas. • Opera con lenguajes formales para obtener nuevos lenguajes y verificar propiedades matemáticas. • Resuelve el problema de generación de lenguajes formales mediante gramáticas apropiadas para diferenciar el alcance de cada elemento de la Jerarquía**.**	

Eje	Actividades	Resultados de Aprendizaje	Grado de Profundidad en el tratamiento
	Trabajo Práctico N° 3: *Teoría de Autómatas* • Identificar los tipos de autómatas (Máquina de Turing, Pila y Finito) y su relación con los tipos de lenguajes que reconocen cada autómata. • Resolver el problema de reconocimiento de lenguajes mediante autómatas (Máquina de Turing, Pila y Finito). • Resolver problemas Turing computables tanto con Máquinas de Turing unicinta como multicintas.	• Identifica los tipos de autómatas para comprender sus limitaciones y alcances. • Resuelve el problema de reconocimiento de lenguajes mediante autómatas. • Resuelve problemas Turing computables tanto con Máquinas de Turing unicinta como multicintas para dimensionar su expresividad como concepto análogo al de algoritmo.	

	Taller: *Diseño de dispositivos para generar un lenguaje de programación*
	• Identificar y definir el lenguaje formal. Para ello:
	  ◦ Especificar el vocabulario.
	  ◦ Identificar los componentes léxicos tales como: identificadores, palabras claves, constantes, etc.
	  ◦ Realizar las expresiones regulares que reconozcan los componentes léxicos.
	• Resolver el problema de la generación del lenguaje. Para ello:
	  ◦ Elaborar la gramática libre de contextos.
		• Identifica problemas que se presentan en el diseño de un lenguaje de programación.
		• Propone soluciones a problemas que se presentan en el diseño de un lenguaje de programación, integrando conceptos y habilidades adquiridas.	

Las actividades planteadas en los trabajos prácticos 1, 2, 3 y Taller: *Diseño de dispositivos para generar un lenguaje de programación*, se corresponden con la formación del eje transversal descripto ya que los estudiantes afianzan en la identificación y formalización de problemas de decisión, identifican problemas tratables e intratables mediante la eficiencia de algoritmos, identifican lenguajes formales y su tipo, resuelven el problema de generación y de reconocimiento de lenguajes formales mediante gramáticas y autómatas.			

Eje	Actividades	Resultados de Aprendizaje	Grado de Profundidad en el tratamiento

Concepción, diseño y desarrollo de proyectos de informática	**Taller**: *Diseño de dispositivos para generar un lenguaje de programación*
	• Diseñar el lenguaje de programación.
	• Especificar los componentes léxicos:
	  ◦ Definir los componentes léxicos.
	  ◦ Realizar las expresiones regulares que reconozcan.
	• Diseñar la gramática libre de contexto que genere el lenguaje
	• Desarrollar la prueba de las expresiones regulares y de la gramática.
		• Diseñe el lenguajes de programación y los dispositivos que permiten su generación.	**Bajo**

Las actividades planteadas se corresponden con la formación del eje transversal descripto ya que los estudiantes realizan el diseño de un pequeño lenguaje de programación, especifican los componentes léxicos y diseñan la gramática libre de contexto. Por lo que, el estudiante realiza actividades que contribuyen al acervo de conocimientos y habilidades para la concepción y diseño de proyectos de informática.			

Eje	Actividades	Resultados de Aprendizaje	Grado de Profundidad en el tratamiento
Utilización de técnicas y herramientas de aplicación en la informática	• Desarrollar programas en JAVA de los: o Algoritmos de Markov que resuelva un problema sencillo (**Trabajo Práctico N° 1**: *Teoría de la Computabilidad y de la Complejidad*). . o Algoritmos de simplificación de gramáticas (**Trabajo Práctico N° 2:** *Teoría de Lenguajes Formales y Gramáticas)*. o Algoritmos de conversión de un autómata finito no determinístico a determinístico y de estados mínimos **(Trabajo Práctico N° 3:** *Teoría de Autómatas*).	• Instale y use el lenguaje de programación JAVA. • Programa los algoritmos y realiza las pruebas unitarias.	**Medio**
	• Realizar las actividades de comprobación de gramáticas y autómatas mediante el uso de JFLAG. Las actividades propuestas son: o Generar hileras para cada lenguaje planteado en la actividad y comprobarlas (**Trabajo Práctico N° 2:** *Teoría de Lenguajes Formales y Gramáticas)*. o Comprobar el funcionamiento de los dispositivos abstracto diseñados en la actividad **(Trabajo Práctico N° 3:** *Teoría de Autómatas*). o Desarrollar las pruebas de la expresiones regulares y de la gramática libre de contexto (T**aller**: *Diseño de dispositivos para generar un lenguaje de programación*.	• Instale y use la herramienta JFLAP. • Verifica la validez de cada una de las soluciones propuestas a los ejercicios planteados. • Interpreta y corrige los errores.	
Las actividades planteadas se corresponden con la formación del eje transversal: *Utilización de técnicas y herramientas de aplicación en la informática*, ya que permite que los estudiantes logren la habilidad para identificar y utilizar los recursos tecnológicos necesarios para la programación y comprobación de las gramáticas y autómatas desarrollados y, capacidad para desenvolverse en entornos de desarrollo de software libre o propietario.			

Eje	Actividades	Resultados de Aprendizaje	Grado de Profundidad en el tratamiento
Fundamentos para el desempeño en equipos de trabajo	**Taller:** Diseño de dispositivos para generar un lenguaje de programación • Trabajar en equipos. Para ello: o Crear el equipo de trabajo. o Realizar los acuerdos iniciales de trabajo (formas y frecuencias de las reuniones). o Planificar las actividades. o Asignar roles, responsabilidades y tareas a cada integrante. o Evaluar el desempeño de los integrantes del equipo de trabajo.	• Organice el grupo de trabajo para realizar las tareas de creación del grupo. • Distribuya responsabilidades y actividades a cada integrante del grupo de trabajo. • Evalué el desempeño de los integrantes y del trabajo en equipo.	**Medio**
Para la realización del taller propuesto como actividades de formación práctica se requiere la formación y trabajo en equipo, por lo que los estudiantes afianzan la competencia de desempeño en equipos de trabajo. Se proponen tareas para la creación, asignación de roles de los integrantes y la autoevaluación del desempeño del grupo.			

Eje	Actividades	Resultados de Aprendizaje	Grado de Profundidad en el tratamiento
Fundamentos para la comunicación efectiva	**Taller:** *Diseño de dispositivos para generar un lenguaje de programación* ● Elaborar el informe técnico del diseño del lenguaje de programación, las expresiones regulares, los autómatas finitos y la gramática libre de contexto. ● Exponer el trabajo realizado en el taller. **Trabajo Práctico N° 1**: *Teoría de la Computabilidad y de la Complejidad* • Elaborar un informe producto de las investigaciones de ejemplos de la clasificación de problemas y de eficiencia de algoritmos.	• Redacta un informe técnico que documente de manera clara, estructurada y coherente el proceso de diseño de un lenguaje de programación y de las actividades de investigación realizadas, asegurando que toda la información relevante esté bien organizada y sea fácilmente comprensible. • Aplica estándares y convenciones comunes en la documentación técnica de las fases de un compilador, asegurando que el informe cumpla con las expectativas de formato, terminología y precisión técnica. • Presenta de manera clara y coherente los resultados del trabajo realizado en el taller, utilizando un lenguaje técnico adecuado. • Exponga el taller organizando la información para facilitar la comprensión a los destinatarios, utilizando recursos gráficos y con técnicas de comunicación oral.	**Medio**
Las actividades de redacción y exposición del trabajo realizado en el taller: *Diseño de dispositivos para generar un lenguaje de programación* y las actividades de investigación descriptas en el trabajo práctico, que requieren un informe por escrito, se corresponde con la formación del eje transversal descripto ya que los estudiantes deben organizar adecuadamente la información, redactar correctamente, utilizar un adecuado estilo de escritura y emplear fuentes de información o referencias bibliográficas acorde a las normas APA. Además, los estudiantes deben exponer el trabajo afianzando la competencia de comunicación oral mediante la utilización correcta de técnicas de comunicación oral y de vocabulario técnico.			

Eje	Actividades	Resultados de Aprendizaje	Grado de Profundidad en el tratamiento
Fundamentos para el aprendizaje continuo	**Trabajo Práctico N° 1**: *Teoría de la Computabilidad y de la Complejidad* • Actividades de investigación: o Investigar en sitios web actualizados sobre la clasificación de problemas y extraer ejemplos para las clases P, NP y NP-completa. o Trabajo de investigación para complementar el estudio de la eficiencia de los algoritmos planteado, mediante la interpretación de papers actuales.	• Investiga y evalúa información de fuentes web actualizadas y confiables, seleccionando ejemplos relevantes y precisos de problemas que pertenecen a las clases P, NP y NP-completa. • Analizar ejemplos de problemas de cada clase, discutiendo las características que los clasifican como miembros de una clase en particular. • Desarrolla habilidades de aprendizaje autónomo, buscando, comprendiendo y contextualizando información nueva sobre la teoría de la complejidad computacional.	**Bajo**
Las actividades planteadas se corresponden con la formación del eje trasversal ya que los estudiantes deben realizar búsquedas de información e investigaciones sobre los temas planteados. Estas actividades promueven en los estudiantes el aprendizaje autónomo, la adaptación a nuevas situaciones y favorece al proceso de aprendizaje a lo largo de toda su vida.			

5.4 Cronograma de formación práctica

Tabla 4: Cronograma para el desarrollo de las Actividades Prácticas

ACTIVIDAD	CARGA HORARIA	CRONOGRAMA DE DESARROLLO
TP N° 1	9 hs	09/4 – 16/04 – 23/4
TP N° 2	12 hs	30/04 – 07/05 – 14/05 – 21/05
TP N° 3	9 hs	04/06 – 11/06 – 18/06 – 25/06
Trabajo de Investigación	6 hs	23/04 – 08/05
Trabajo de Articulación – Mundo Real	9 hs	29/ 05 - 04/06 – 11/06
TOTAL	**45 hs**	

6. BIBLIOGRAFÍA

6.1. BIBLIOGRAFÍA GENERAL Y ESPECÍFICA

TÍTULO	AUTORES	EDITORIAL	EJEMPLARES DISPONIBLES	AÑO DE EDICIÓN
Lógica Matemática y Computabilidad	San Roman, Sancho	Díaz De Santos S. A.	1	1990
Teoría de Lenguajes, Gramáticas y Autómatas	Alfonseca Manuel; Sancho, Justo y Martínez Orga	Ediciones Universidad y Cultura	1	1990
Introduction to Fornal Languages Theory.	Michael A. Harrison	Addison-Wesley Publishing	1	1978
Compiladores. Principios, técnicas y herramientas	Alfred V. Aho, Monica S. Lam, Ravi Sethi y Jeffrey D. Ullman	Pearson Education, Inc	1	2008
Algorithms and Complexity	Wilf, Herbert S.	Taylor & Francis	1	2002
Construcción de Compiladores - Principios y Práctica	LAUDON, Kenneth C y LAUDON, Jane Price	México Thomson	1	2005
Compiladores: Teoría e Implementación	RUIZ CATALAN, Jacinto	México Alfaomega	1	2010
Implementing Programming Languages: An introduction to compilers and interpreters	RANTA, Aarne	London College Publications	1	2012

La bibliografía referenciada se encuentra en Biblioteca-Central de la UNSE o en Centro de Documentación-Dpto. de Informática.

6.2. DOCUMENTOS Y SITIOS WEB

Parberry, Ian.	*Lecture Notes on Algorithm Analysis and Computational Complexity*. 2001. Disponible en URL: https://everythingcomputerscience.com/books/PDFs/algorithmAnalysis2.pdf
Amalia Duch	Análisis de Algoritmos. Barcelona, marzo de 2007. Disponible en: https://www.cs.upc.edu/~duch/home/duch/analisis.pdf
Rosa Guerequeta y Antonio Vallecillo	Técnicas de diseño de algoritmos. Universidad de Málaga. 2000. Disponible en: https://ia802903.us.archive.org/16/items/TecnicasDiseoAlgoritmosGuequeretaYVallecillo/T%C3%A9cnicas%20de%20Dise%C3%B1o%20de%20Algoritmos%20-%20Guequereta%20y%20Vallecillo.pdf
Alvarez Margarita y Budán Paola	Fundamentos Teóricos de la Ciencia de la Computación.. 2022. Disponible en: https://cuv-fceyt.unse.edu.ar/course/view.php?id=2699

7. ESTRATEGÍAS METODOLÓGICAS

7.1. ASPECTOS PEDAGÓGICOS Y DIDÁCTICOS

Considerando que la cátedra concibe la educación no como una serie de acciones donde el docente es el único protagonista que trasmite y el único responsable del aprendizaje, por el contrario, es el alumno el que debe ser co-responsable y protagonista activo de su aprendizaje, es que se seleccionaron las siguientes técnicas metodológicas para poner en juego:

- En las clases teóricas
  - En la primera clase se realizará una presentación de la asignatura, indicando los objetivos de aprendizaje que persigue la misma, las unidades temáticas, los requisitos para regularizar y las actividades prácticas, a fin de que el alumnado tome conocimiento de la asignatura.
  - Al comienzo de cada unidad temática se presentarán los contenidos que se desarrollarán, su importancia y usos. De igual forma al finalizar las unidades temáticas se presentará un mapa conceptual con los conceptos involucrados de forma tal de lograr una integración y mayor comprensión de los conceptos.
  - Los temas teóricos se presentarán utilizando PowerPoint, con gráficos, tablas y ejemplos ilustrativos.
  - Tanto en las clases prácticas como en las teóricas, se buscará incentivar una activa participación de los estudiantes planteándoles interrogantes, solicitándoles ejemplos, y fundamentalmente proponiendo actividades para ser resueltas en forma grupal, dándoles el tiempo necesario para la reflexión y la asimilación de los conceptos.
- En las clases prácticas
  - La técnica metodológica por excelencia será el trabajo grupal que permite promover la construcción compartida del conocimiento y lograr así no sólo la apropiación activa del mismo por parte de los miembros del grupo, sino también la indispensable socialización del estudiante, ya que toda su vida deberá transcurrir en contacto y en cooperación con sus semejantes.
- Aula Virtual
  - Se dispone de un aula virtual bajo la plataforma Moodle. En ella se encuentra el material de la teoría, las actividades prácticas y bibliografía. Además, permite la comunicación docente-alumno y entre alumnos a través de foros y correo electrónico.

La articulación e integración teoría y práctica supone la alternancia, sucesión y predominio de momentos teóricos y prácticos. Se incluyen ejercitación, ejemplificación y resolución de problemas de modo de integrar los conceptos con la praxis.

7.3- MECANISMOS PARA LA INTEGRACIÓN DE DOCENTES

Se realizarán actividades de revisión y coordinación en el Trayecto de Ciencias Básicas y Específicas en el marco de la Comisión de Seguimiento Académico. Además, se realizarán reuniones periódicas con las cátedras del mismo año con el fin de aunar tareas conjuntas de integración.

7.4- RECURSOS DIDÁCTICOS

- Tiza, pizarrón, computadora y proyector. Estos se utilizarán para las clases expositivas del docente.
- Software libre: JFLAG.

8. EVALUACIÓN

8.1. EVALUACIÓN DIAGNÓSTICA

No está prevista, sin embargo, la primera unidad cumple el doble rol de constituir un elemento motivador para incursionar en los fundamentos de la ciencia de la computación y, por otra parte, permite a los docentes disponer de un instrumento para diagnosticar la capacidad de los alumnos para diseñar algoritmos y programas.

8.2. EVALUACIÓN FORMATIVA: es de carácter continuo y está dirigida a evaluar el proceso de enseñanza-aprendizaje, por lo que se llevará a cabo durante todo el desarrollo de la asignatura. En el marco de esta formación continua, los resultados de aprendizaje se evaluarán de acuerdo con los criterios de evaluación que se consignan en la siguiente Tabla.

Resultados de Aprendizaje	Criterio de Evaluación
Identifica problemas de decisión para consolidar conceptos abstractos de clasificación de problemas.	Identifica correctamente los problemas de decisión.
Formaliza problemas de decisión para afianzar sus habilidades de expresión matemática –lógica.	Formaliza correcta los problemas de decisión consignando con exactitud todos los elementos de la cuádrupla.
Identifica problemas tratables e intratables mediante la eficiencia de algoritmos.	Identifica correctamente problemas tratables e intratables mediante el cálculo de la eficiencia de algoritmos.
Identifica lenguajes formales para poder diferenciar sus características según la expresividad de cada uno.	Identifica correctamente los tipos de lenguajes.
Opera con hileras para obtener nuevos hileras y verificar propiedades matemáticas.	Opera correctamente con hileras.
Opera con lenguajes formales para obtener nuevos lenguajes y verificar propiedades matemáticas	Opera correctamente con lenguajes.
Resuelve el problema de generación de lenguajes formales mediante gramáticas apropiadas para diferenciar el alcance de cada elemento de la Jerarquía.	Diseña correctamente las gramáticas de forma que generen lenguajes de acuerdo a su tipo. Prueba la generación de hileras mediante el proceso de derivación y árboles sintácticos.
Resolver el problema de reconocimiento de lenguajes mediante autómatas.	Diseña y prueba el correcto funcionamiento de los autómatas de forma de que reconozcan lenguajes de acuerdo a su tipo.
Resuelve problemas Turing computables tanto con Máquinas de Turing unicinta como multicintas para dimensionar su expresividad como concepto análogo al de algoritmo.	Diseña correctamente las máquinas de Turing que computen funciones. Prueba las máquinas de Turing con varios ejemplos.
Identifica problemas que se presentan en el diseño de un lenguaje de programación.	Identifica correctamente los problemas que se presentan al diseñar lenguajes.
Propone soluciones a problemas que se presentan en el diseño de un lenguaje de programación, integrando conceptos y habilidades adquiridas.	Las soluciones se evidencian claramente en el lenguaje diseñado.
Diseñe el lenguajes de programación y los dispositivos que permiten su generación.	Diseña correctamente el lenguaje, cumple con todas las especificaciones previstas en la metodología y está perfectamente documentado.

Instale y use el lenguaje de programación JAVA.	Instala Java resolviendo las situaciones excepcionales que se le pudieran presentar durante la instalación.
Programa los algoritmos y realiza las pruebas unitarias.	Demuestra dominio en programación y prueba de programas.
Instale y use la herramienta JFLAP.	Instala JFLAP resolviendo las situaciones excepcionales que se le pudieran presentar durante la instalación.
Verifica la validez de cada una de las soluciones propuestas a los ejercicios planteados. Interpreta y corrige los errores	Demuestra dominio en el uso de JFLAP para la verificación y corrección de errores

8.3. EVALUACIÓN PARCIAL

8.3.1. PROGRAMA DE EVALUACIONES PARCIALES

Se realizará dos parciales prácticos con sus correspondientes recuperatorios y parciales teóricos.

EVALUACIÓN PRÁCTICA PARCIAL Nº 1

ESTRUCTURA	OBJETIVO	CRITERIOS DE EVALUACIÓN
Ejercicio para formalizar problemas y determinar la eficiencia de un algoritmo.	Determinar si los alumnos han adquirido habilidad para formalizar problemas y capacidad para determinar la eficiencia de los algoritmos.	• Formalización correcta de problemas. • Adecuada determinación de la eficiencia de los algoritmos.
Ejercicio para determinar tipo de lenguaje y elaboración de la gramática.	Determinar si los alumnos han adquirido habilidad para generar y reconocer gramáticas y lenguajes de distinto tipo.	• Correctitud en el reconocimiento de lenguajes de distinto tipo. • Correctitud en la generación de gramáticas de distinto tipo
Ejercicio para aplicar métodos desarrollados en la unidad 3.	Determinar si los alumnos son capaces de aplicar los algoritmos estudiados.	• Aplicación correcta de los algoritmos estudiados

EVALUACIÓN PRÁCTICA PARCIAL Nº 2

ESTRUCTURA	OBJETIVO	CRITERIOS DE EVALUACIÓN
Ejercicio para aplicar métodos desarrollados en la unidad 4.	Determinar si los alumnos son capaces de aplicar los algoritmos estudiados.	• Aplicación correcta de los algoritmos estudiados.
Dos ejercicios para construir autómatas de distinto tipo	Determinar si los alumnos son capaces de construir autómatas.	• Correctitud en la construcción de autómatas de distintos tipos.

Parciales Teóricos: Preguntas Multi-Choice

CRONOGRAMA DE EVALUACIONES TEÓRICAS

FECHAS	TEMAS
Viernes 19/04	Parcial 1: Teoría de la Computabilidad - Teoría de la complejidad - Eficiencia de algoritmos. Clasificación de problemas
Viernes 24/05	Parcial 2: Teoría de lenguajes, Gramáticas y Características de la gramática

Viernes 28/06	Parcial 3: Autómata Finito, Autómata Pila y Máquina de Turing

CRONOGRAMA DE EVALUACIONES PRÁCTICAS

Martes 28/05	Parcial 1
Martes 18/06	Parcial 2
Martes 02/07	Recuperatorio Integral – Entrega articulación

8.3.2- CRITERIOS DE EVALUACIÓN

Los criterios de evaluación de los parciales teóricos son:

El estudiante debe ser capaz de:

Parcial 1: Teoría de la Computabilidad

1. Reconocer y comprender correctamente los conceptos constructivos de la teoría de la computación.
2. Comprender con claridad el concepto de problema, su formalización y viabilidad.

Parcial 2: Teoría de la complejidad - Eficiencia de algoritmos. Clasificación de problemas

1. Comprender correctamente la teoría de la complejidad
2. Identificar adecuadamente los tipos de análisis de algoritmos
3. Comprender correctamente el orden de un algoritmo.
4. Interpretar correctamente la eficiencia asintótica.
5. Reconocer adecuadamente las propiedades de las clases de problemas.
6. Comprender la importancia de la clasificación de problemas.

Parcial 3: Teoría de lenguajes, Gramáticas y Características de la gramática

1. Conocer detalladamente los conceptos básicos de la teoría de lenguajes formales.
2. Distinguir correctamente los tipos de lenguajes y gramáticas.
3. Interpretar correctamente los algoritmos de eliminación de recursividad a izquierda, factorización a izquierda y de los algoritmos para convertir la gramática en gramática propia.
4. Identificar adecuadamente los operadores para realizar las expresiones regulares

Parcial 4: Autómata Finito y Autómata Pila

1. Conocer con precisión los conceptos involucrados en los Autómatas finitos y de Pila.
2. Interpretar adecuadamente los algoritmos para pasar de un autómata finito no determinístico a determinístico y obtener un autómata finito de estados mínimos

Parcial 5: Máquina de Turing

1. Conocer con precisión los conceptos involucrados en la Máquina de Turing
2. Distinguir adecuadamente las distintas Interpretaciones de la Máquina de Turing
3. Comprender correctamente la importancia la Máquina de Turing

Los criterios de evaluación de los parciales de práctica están consignados en el apartado 8.3.1. Los criterios de evaluación del taller están especificados en el ítem correspondiente de la descripción del mismo y en el apartado 5.2.-Criterios de evaluación generales. Además, en el apartado: Evaluación Formativa se consignan los criterios de evaluación para la formación en los ejes transversales.

8.3.3- ESCALA DE VALORACIÓN

La escala de valoración a emplear: para las evaluaciones parciales y los recuperatorios será cuantitativa: (Escala: 1 al 100). El puntaje mínimo para aprobar los parciales es de **cincuenta (50) puntos (sobre una**

calificación máxima de 100). Se otorga una sola recuperación en caso de desaprobación. Si se obtiene como **mínimo setenta (70) puntos** se puede acceder al **Sistema de Promoción.**

8.4- EVALUACIÓN INTEGRADORA

No está prevista.

8.5. EVALUACIÓN SUMATIVA

8.5.1- CONDICIONES PARA LOGRAR LA PROMOCIÓN SIN EXAMEN FINAL DE LA ASIGNATURA.

a) Asistir como mínimo al 80 % del total de sesiones.

b) Aprobar las evaluaciones prácticas parciales con un puntaje ≥70 puntos

c) Aprobar las evaluaciones teóricas con un puntaje ≥70 puntos

d) Aprobar la actividad de resolución de problemas del mundo real con un puntaje ≥70 puntos.

8.5.2- CONDICIONES PARA LOGRAR LA REGULARIDAD DE LA ASIGNATURA

a) Asistir como mínimo al 80 % del total de sesiones.

b) Aprobar las evaluaciones prácticas parciales o los recuperatorios.

c) Aprobar la actividad de resolución de problemas del mundo real.

8.6. EXAMEN FINAL

La evaluación final será escrita u oral sobre los temas incluidos en la programación de la asignatura.

8.7. EXAMEN LIBRE

Prerrequisitos: Deberán aprobar las evaluaciones correspondientes a las siguientes etapas y subetapas cada una de ellas eliminatorias:

Primera etapa: presentación de la actividad de resolución del mundo real, para lo cual debe solicitar el tema de la actividad por lo menos 10 días antes del examen. Este trabajo debe ser presentado 5 días antes del examen.

Segunda etapa:.

a. Temas a desarrollar, ejemplos y problemas correspondientes a las Unidades 1 y 2. Duración 4 hs. Modalidad escrita.

b. Temas a desarrollar, ejemplos y problemas correspondientes a las Unidades 3 y 4. Duración 4 hs. Modalidad escrita.

Duración total 8 (ocho) hs. repartidas en 2 días.

Tercera etapa: Evaluación oral. Se utilizará la misma modalidad que se utiliza para los alumnos regulares.

…….

Msc. Ing. Margarita María Álvarez