

UNIVERSIDAD NACIONAL DE SANTIAGO DEL ESTERO
FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍAS

PLANIFICACIÓN ANUAL 2023

Asignatura: Lenguajes de Programación y Compiladores

**LICENCIATURA EN SISTEMAS DE
INFORMACIÓN**
**PLAN DE ESTUDIOS: 2011 – Innovación Curricular
2022**

EQUIPO CÁTEDRA

Profesor Asociado: Msc. Ing. Margarita María Álvarez

JTP: Dra. Paola Budán

1. IDENTIFICACIÓN

1.1. NOMBRE DE LA ASIGNATURA: LENGUAJES DE PROGRAMACIÓN Y COMPILADORES

1.2. CARRERA: LICENCIATURA EN SISTEMAS DE INFORMACIÓN

1.3. PLAN DE ESTUDIOS: Plan 2011- Innovación curricular 2022

1.4. AÑO ACADÉMICO: 2023

1.5. CARÁCTER: Obligatorio

1.6. UBICACIÓN DE LA ASIGNATURA EN EL PLAN DE ESTUDIOS:

1.6.1. MÓDULO: 8° - **AÑO:** 4°

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 III (REGULAR)
- TEORÍA DE LA COMPUTACIÓN (APROBADA)

1.6.3.2. Posteriores:

- PROGRAMACIÓN AVANZADA

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 y laboratorio de informática de la FCEyT.

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 abordan conceptos de los lenguajes de programación y se discuten aspectos referidos al diseño y a la implementación de los mismos, de modo tal que le permita al estudiante aprovechar su potencial y permitir lograr un uso eficiente de los lenguajes de programación.

Otro gran tópico que trata la asignatura es la aplicación de las gramáticas regulares y libres de contexto y, de los autómatas finitos y de pila en los compiladores e intérpretes. Se proporciona la justificación teórica y práctica de los instrumentos necesarios para la construcción de los compiladores.

Los conceptos y técnicas correspondientes al tema Compiladores le permitirán al alumno desarrollar un intérprete sencillo y además, pueden ser utilizadas en el desarrollo de software de aplicación general.

El valor formativo de todos los conceptos desarrollados en la asignatura es esencial en el diseño de Compiladores e Intérpretes y además, para proveer una base sólida para el cabal ejercicio profesional en el desarrollo de programas.

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 Programación y Teoría de la Computación.

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.
- Gestionar, planificar, ejecutar y controlar 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.

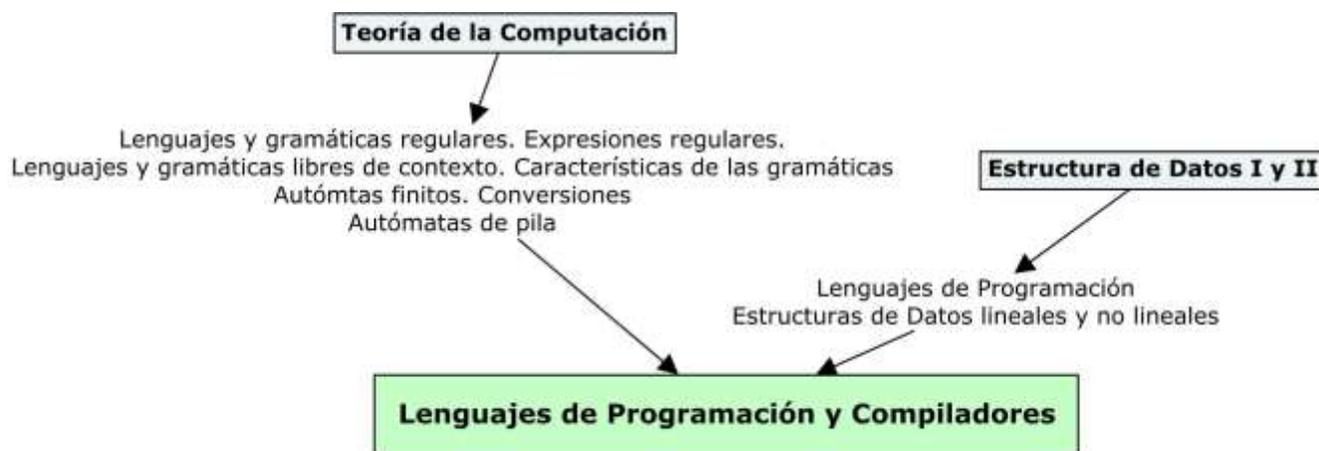


Figura 1. Integración horizontal y vertical con otras asignaturas

3. OBJETIVOS

- a. Conocer los diferentes conceptos de los lenguajes de programación.
- b. Diferenciar los criterios de diseño y de implementación de lenguajes de programación para ser capaces de justificar una elección.

- c. Construir compiladores/intérpretes sencillos mediante el desarrollo de las fases del proceso de traducción y con el empleo de herramientas generadoras de analizadores léxicos y sintácticos.
- d. Aplicar procedimientos que permitan la generación de código intermedio y su optimización.

4. SELECCIÓN Y ORGANIZACIÓN DE CONTENIDOS

4.1 CONTENIDOS MÍNIMOS ESTABLECIDOS EN EL PLAN DE ESTUDIOS

Lenguajes de Programación: Entidades y ligaduras. Criterios de diseño y de implementación de lenguajes de programación. Aplicación de las gramáticas regulares y libres de contexto y, de los autómatas finitos y de pila en los compiladores e intérpretes. Estructura. Analizadores léxico, sintáctico, traducción dirigida por la sintaxis, definición dirigida por la sintaxis. Analizador semántico. Sistema de Tipos. Comprobación de tipos. Tabla de símbolos. Generación y optimización de código intermedio. Generación de Código Objeto.

4.2. PROGRAMA SINTÉTICO SOBRE LA BASE DE LOS CONTENIDOS MÍNIMOS

LENGUAJES DE PROGRAMACIÓN: Entidades, ligaduras y variables. Sistema de tipos. Criterios de diseño y de implementación de lenguajes de programación.

COMPILADORES:

Traductores: conceptos, ensambladores, compiladores e intérpretes. Fases de un compilador.

Analizador lexicográfico: conceptos y funciones. Especificación y reconocimiento de tokens.

Analizador sintáctico: funciones. Analizadores sintácticos descendentes y ascendentes.

Analizador semántico. Traducción dirigida por la sintaxis. Comprobación de tipos.

Generación de código intermedio: funciones. Distinto tipo de representaciones.

Optimización y generación de código objeto.

4.3 ARTICULACIÓN TEMÁTICA DE LA ASIGNATURA

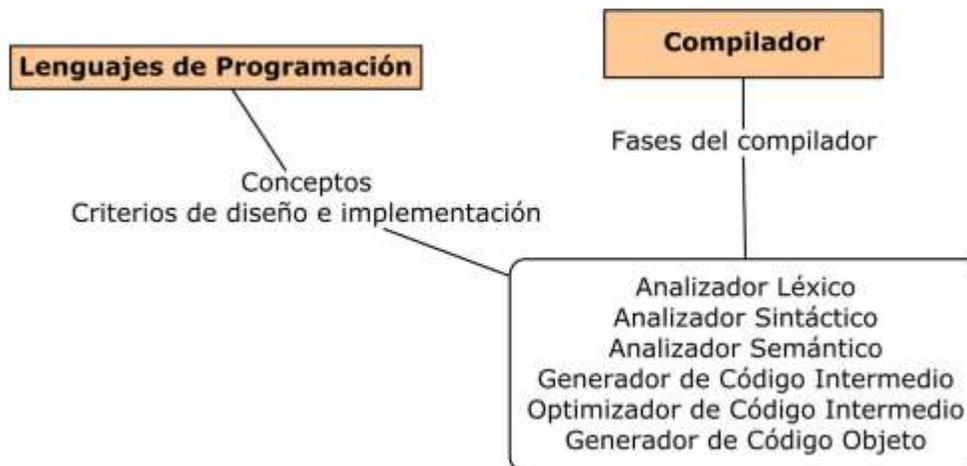


Figura 1. Articulación temática de la asignatura

4.4. PROGRAMA ANALÍTICO

Unidad 1: LENGUAJES DE PROGRAMACIÓN

Entidades, Atributos y Ligaduras. Tiempos de Ligadura: ligadura estática y dinámica. Variables: declaraciones de tipo, ligadura de tipo; tiempo de vida de la variable; alcance, visibilidad y ambiente de referenciamiento.

Sistemas de tipo: tipo y tiempo de chequeo. Lenguajes fuertemente tipados. Equivalencia de tipo.

Criterios de diseño y de implementación de lenguajes de programación.

Unidad 2: COMPILADORES

Tipo de traductores: Ensambladores, compiladores e intérpretes. Características de cada uno. Comparación. Análisis y síntesis de la compilación. Herramientas para la construcción de compiladores.

Unidad 3: ANÁLISIS LÉXICO

Funciones principales y secundarias. Componentes léxico, patrones, lexemas y atributos. Manejo de buffers. Especificación y reconocimiento de tokens. Definiciones regulares. Diagrama de transición. Generador automático de analizadores léxicos.

Unidad 4: ANÁLISIS SINTÁCTICO

Funciones. Métodos descendentes y ascendentes. Manejo de errores sintácticos. Estrategia. Analizador sintáctico descendente: por descenso recursivo, analizador sintáctico predictivo no recursivo. Gramáticas LR(k). Implantación de un analizador sintáctico ascendente mediante poda de mangos. Conflictos en el análisis. Analizadores Sintácticos LR. Generador automático de analizadores sintácticos.

Unidad 5: ANÁLISIS SEMÁNTICO

Traducción dirigida por la sintaxis. Definición dirigida por la sintaxis. Reglas semánticas. Esquema de traducción. Acciones semánticas. Comprobación de tipos.

Unidad 6: FASES DE SÍNTESIS

Generación de código Intermedio. Distintos tipos de representaciones: Árbol sintáctico para expresiones, Grafos dirigidos acíclicos para expresiones (GDA), Notación postfija y Código de tres direcciones. Implementaciones de proposiciones de tres direcciones. Optimización de Código Intermedio. Generación de Código Objeto. Tabla de símbolos.

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
Introducción a los Compiladores	2 hs.	1° Semana
Análisis Léxico	4 hs.	2° Semana
		3° Semana
Análisis Sintáctico	8 hs.	4° Semana
		5° Semana
		6° Semana
		7° Semana
Análisis Semántico	4 hs.	8° Semana
		9° Semana
Fases de Síntesis	4 hs.	10° Semana
		11° Semana
Conceptos de Lenguajes de Programación	4 hs.	12° Semana
		13° Semana

5. FORMACIÓN PRÁCTICA

5.1. DESCRIPCIÓN DE LAS ACTIVIDADES DE FORMACIÓN PRÁCTICA

Durante el cursado los alumnos trabajan con trabajos prácticos y un taller para desarrollar un intérprete. Los trabajos prácticos poseen ejercicios tipos de los diferentes temas que se abordan en la teoría, y otros ejercicios en donde el alumno tiene que desarrollar habilidades de programación y trabajo en equipo, que se complementan con lecturas de papers actualizados sobre el tema del trabajo práctico. Los trabajos prácticos están diseñados de manera de cumplir con la resolución de problemas específicos de la informática, en el dominio particular de las herramientas de análisis lexicográfico y sintáctico, a la vez que se complementan con la realización de proyectos plasmadas en actividades en las cuales los alumnos deben programar. En el trabajo práctico N° 1 se desarrollan manualmente analizadores léxicos, se instala ANTLR, se programa un analizador léxico en JAVA y se comienzan con ejercicios prácticos en ANTLR, profundizando en el orden en el cual se consignan las expresiones regulares como método para que el alumno comprenda mejor los errores en este tipo de analizador. Mientras realizan el trabajo práctico, los alumnos deben construir una WIKI colaborativa en la plataforma sobre alguna investigación de actualidad, por ejemplo, sobre las técnicas que se usan actualmente en los compiladores. En la parte I del trabajo práctico N° 2, se desarrollan varios ejercicios de analizadores sintácticos predictivos no recursivos y familias de analizadores LR. En la parte II se realizan ejercicios sencillos de traducción de expresiones a notación postfija, árbol sintáctico y GDA, código de 3 direcciones, triplas y cuádruplas.

Además, los alumnos desarrollan el taller denominado *Diseño y construcción de un intérprete*, que se entiende como una actividad de resolución de problemas del mundo real. Los contenidos más importantes que abarca este taller son:

- 1) Analizador léxico. Generador de AL.
- 2) Analizador sintáctico. Generador de AS.
- 3) Analizador semántico.
- 4) Intérprete.

Los *objetivos de este taller* son: (i) Introducir al alumno en los problemas concretos que se enfrentan durante el desarrollo de programas traductores; (ii) Reforzar, desde el punto de vista práctico, los conocimientos adquiridos sobre analizadores lexicográficos, sintácticos y estructura de programas traductores; y (iii) Adquirir habilidades para construir compiladores/intérpretes sencillos.

La *Metodología* a seguir para desarrollar el taller es:

1. Estudiar los temas que comprende el trabajo
2. Especificar el vocabulario
3. Describir identificadores, palabras claves, constantes, etc.
4. Elaborar la gramática y probar manualmente la misma.
5. Con respecto al Analizador Lexicográfico:
 - a. Elaborar las Expresiones Regulares
 - b. Codificarlo mediante el empleo del generador automático de analizadores léxicos como ANTLR.
 - c. Probar que funcione correctamente.
6. Con respecto al Analizador Sintáctico:
 - a. Codificarlo mediante el empleo del generador automático, como por ejemplo, usando ANTLR.
 - b. Probar que funcione correctamente.

El taller se *evalúa* con un puntaje máximo 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**.

Los criterios que se utilizan para la evaluación son:

1. Entrega del informe en tiempo y forma: se utilizará la siguiente criterios de evaluación:
 - a. Organización de la información y pertinencia de los temas incluidos.
 - b. Correcta redacción y uso adecuado del estilo de escritura que facilite la comprensión.

- c. Utilización de ejemplos acordes al tema en cuestión
 - d. Inclusión de fuentes de información o referencias acorde a las normas APA.
2. Funcionamiento del intérprete: se evaluará con los siguientes criterios:
 - a) Completitud de las operaciones exigidas.
 - b) Funcionamiento correctamente.
 - c) Detección de errores y emisión de los mensajes correspondientes.
 - d) Facilidad de uso.
 - e) Diseños de pantalla adecuados y recursos gráficos convenientes.
 3. 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.
 - c) Utilización de vocabulario técnico.
 4. Trabajo en grupo y planificación y gestión del proyecto: se evaluarán de acuerdo a los siguientes criterios de evaluación:
 - a) Realización de los acuerdos iniciales de trabajo (formas y frecuencias de las reuniones).
 - b) Realización de la planificación del proyecto identificando actividades y recursos de desarrollo, definición de la calendarización y especificación de roles de cada integrante.

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.

Durante la ejecución del taller se propiciará el trabajo en equipo y durante la evaluación del taller, se fomentará la comunicación efectiva.

Cronograma

La presentación del trabajo se realizará en el mes de noviembre.

5.2. FORMACIÓN EN EJES TRANSVERSALES

En la tabla siguiente se relaciona para cada uno de los ejes transversales las actividades prácticas seleccionadas con los resultados de aprendizaje.

Tabla 3: Formación en Ejes Transversales

Eje	Actividades	Resultados de Aprendizaje	Grado de Profundidad en el tratamiento
Identificación, formulación y resolución de problemas de informática	<ul style="list-style-type: none"> • Trabajos prácticos • Taller: Diseño y construcción de un intérprete 	<ul style="list-style-type: none"> • Identificar componentes y funcionalidades de los analizadores. • Resolver problemas de análisis léxico y sintáctico. • Resolver problemas de código intermedio y optimización de código. • Identificar y resolver los problemas que se presentan en la construcción de un intérprete. 	Alto
Concepción, diseño y	<ul style="list-style-type: none"> • Taller: Diseño y construcción de 	<ul style="list-style-type: none"> • Diseñar y desarrollar el intérprete. • Realizar pruebas de 	Medio

Eje	Actividades	Resultados de Aprendizaje	Grado de Profundidad en el tratamiento
desarrollo de proyectos de informática	un intérprete	funcionamiento. <ul style="list-style-type: none"> Planificar y gestionar el proyecto de diseño y construcción de un intérprete. 	
Utilización de técnicas y herramientas de aplicación en la informática	<ul style="list-style-type: none"> Actividades de programación en los trabajos prácticos Taller: Diseño y construcción de un intérprete 	<ul style="list-style-type: none"> Instalar y usar la herramienta de generación de analizadores: ANTLR. Aplicar las técnicas y herramientas para crear los analizadores. Interpretar los errores. 	Medio
Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.	<ul style="list-style-type: none"> Taller: Diseño y construcción de un intérprete 	Diseñar y desarrollar un intérprete sobre temas / lenguajes actuales.	Alto
Fundamentos para el desempeño en equipos de trabajo	<ul style="list-style-type: none"> Taller: Diseño y construcción de un intérprete 	<ul style="list-style-type: none"> Crear el equipo de trabajo. Planificar las actividades. Asignar roles, responsabilidades y tareas a cada integrante. Evaluar el desempeño de los integrantes del equipo de trabajo. 	Medio
Fundamentos para la comunicación efectiva	<ul style="list-style-type: none"> Taller: Diseño y construcción de un intérprete. 	<ul style="list-style-type: none"> Elaborar el informe técnico del taller, organizando la información adecuadamente de forma tal de facilitar la comprensión a los destinatarios, utilizando los recursos gráficos y los medios necesarios para comunicar de forma efectiva la información, con adecuado uso de las normas gramaticales y con bibliografía actualizada y debidamente referenciada. Exponer el taller organizando la información para facilitar la comprensión a los destinatarios, utilizando recursos gráficos y con técnicas de comunicación oral. 	Medio
Fundamentos para el aprendizaje continuo	<ul style="list-style-type: none"> Actividades de lecturas de papers. 	Interpretar los conceptos especificados en los papers.	Bajo

5.3 CRONOGRAMA DE FORMACIÓN PRÁCTICA

ACTIVIDAD	CARGA HORARIA	CRONOGRAMA DE DESARROLLO
Trabajo Práctico 1 – Analizador Léxico e Introducción a ANTLR	9 hs	1º, 2º y 3º Semana (se prevén 2 horas del total para el desarrollo del AL del

ACTIVIDAD	CARGA HORARIA	CRONOGRAMA DE DESARROLLO
		taller)
Trabajo Práctico 2 (parte I) – Analizador Sintáctico	9 hs	4°, 5° y 6° Semana (se prevén 2 horas del total para el desarrollo del AS del taller)
Trabajo Práctico 2 (parte II) – Generación y Optimización de código	9 hs	7°, 8° y 9° Semana (se prevén 2 horas del total para el desarrollo del A Semántico del taller)
Taller: Diseño y construcción de un intérprete.	12 hs. (6 horas considerando las semanas anteriores y 6 horas de consulta específica)	Es una actividad <i>transversal</i> que se desarrolla <i>durante todas las semanas de práctica</i> . Fechas de consultas específicas: 11° y 12° semana.
Parcial de Práctica	3 hs	13° semana
Recuperatorio Parcial	3 hs	14° semana
Presentación del Taller	3 hs	13° semana
Defensa del Taller	3 hs	15° semana

6. BIBLIOGRAFÍA

TÍTULO	AUTORES	EDITORIAL	EJEMPLARES DISPONIBLES	AÑO DE EDICIÓN
Compiladores. Principios, técnicas y herramientas	Alfred V. Aho, Monica S. Lam, Ravi Sethi y Jeffrey D. Ullman	Pearson Education, Inc	1	2008
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
Programming language concepts-Third edition	Carlo Ghezzi y Mehdi Jazayeri	John Wiley & Sons	1	1996
Concepts Programming Language	Robert Sebesta	Pearson	1	2012

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

7. ESTRATEGÍAS METODOLÓGICAS

7.1. ASPECTOS PEDAGÓGICOS Y DIDÁCTICOS

Se pretende que el aula sea un espacio de diálogo y trabajo constructivo donde tanto los alumnos como los docentes interactúen permanentemente y que estén dispuestos para el interaprendizaje y para entreayudarnos. Considerando que la cátedra concibe la educación no como una serie de acciones donde el docente es el único protagonista que transmite 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 de formación 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.
 - Para la formación práctica se emplea principalmente la estrategia didáctica de Aprendizaje basada en proyectos.

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.

La formación en los ejes transversales se desarrollará en los trabajos prácticos y en el taller: Diseño y construcción de un intérprete.

7.2- MECANISMOS PARA LA INTEGRACIÓN DE DOCENTES

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

Por otra parte, en la asignatura se prevé el desarrollo de un taller, cuyo objetivo es la construcción de un intérprete para un lenguaje de programación diseñado por los alumnos en la asignatura Teoría de la Computación.

7.3- RECURSOS DIDÁCTICOS

Se utilizarán como recursos didácticos:

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

- Software: generadores de analizadores léxicos y sintácticos. Estos se utilizarán para realizar ejercicios propuestos en los trabajos prácticos y para la elaboración del intérprete propuesto en la actividad de Resolución de problemas del mundo real propuesta.

8. EVALUACIÓN

8.1. EVALUACIÓN DIAGNÓSTICA

Se realizará una evaluación diagnóstica con ejercicios de la teoría de lenguajes formales y gramáticas y la teoría de autómatas.

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.

8.3. EVALUACIONES PARCIALES

8.3.1. PROGRAMA DE EVALUACIONES PARCIALES

Evaluación Parcial Práctica: se llevará a cabo una evaluación parcial práctica con ejercicios de los temas incluidos en todas las unidades.

Parciales Teóricos: temas incluidos en todas las unidades.

CRONOGRAMA DE EVALUACIONES PARCIALES DE TEORÍA

TEMAS	FECHA
Evaluación 1: Introducción a la Construcción de Compiladores y Analizador Léxico	4° semana
Evaluación 2: Analizador Sintáctico	6° semana
Evaluación 3: Conceptos de lenguajes, Analizador Semántico y Traducción dirigida por la sintaxis	10° semana
Evaluación 4: Generación de Código intermedio, Optimización de Código Intermedio, Generación de código Objeto y tabla de símbolos.	12° semana

CRONOGRAMA DE EVALUACIONES DE PRÁCTICA

Evaluación	Fecha
Evaluación Parcial: Los analizadores sintácticos. Generación y optimización de código	11° semana
Evaluación Recuperatoria: Los analizadores sintácticos. Generación y optimización de código	13° semana
Defensa del Taller: exposición oral y ejemplificación del funcionamiento de los analizadores.	14° y 15° semana

8.3.2- CRITERIOS DE EVALUACIÓN

Los criterios de evaluación de los parciales teóricos son:

El estudiante debe ser capaz de:

1. Reconocer y comprender las fases de un compilador.
2. Interpretar correctamente el significado de lexemas, componentes léxicos y patrones.
3. Comprender la especificación y reconocimiento de componentes léxicos.

4. Conocer la clasificación de los analizadores sintácticos e interpretar adecuadamente los algoritmos de análisis sintáctico.
5. Identificar las funciones del analizador semántico y comprender los formalismos de traducción dirigida por la sintaxis.
6. Comprender las distintas representaciones intermedias y reconocer las técnicas de optimización de código.
7. Identificar las funciones del generador de código objeto.
8. Conocer las estructuras usadas en la tabla de símbolo.
9. Explicar los criterios de diseño e interpretación de los lenguajes de programación e identificar los conceptos constructivos de los lenguajes de programación.

Los criterios de evaluación a aplicar en las evaluaciones parciales y en los recuperatorios son los que se detallan a continuación.

El estudiante debe ser capaz de:

- Interpretar y aplicar correctamente los algoritmos de análisis sintáctico.
- Traducir un código a una representación intermedia.
- Realizar especificaciones de componentes léxicos.

8.3.3- ESCALA DE VALORACIÓN

La escala de valoración a emplear: para las evaluaciones parciales y los recuperatorios será cuantitativa politómica: (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 con Coloquio Final**.

8.4- EVALUACIÓN INTEGRADORA

No se prevé

8.5. EVALUACIÓN SUMATIVA

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

- a) Asistir como mínimo al 75 % del total de sesiones.
- b) Aprobar la evaluación práctica parcial con un puntaje ≥ 70 puntos
- c) Aprobar las evaluaciones teóricas con un puntaje ≥ 70 puntos
- d) Aprobar el Taller con un puntaje ≥ 70 puntos.

8.5.2- CONDICIONES PARA LOGRAR LA REGULARIDAD DE LA ASIGNATURA

- a) Asistir como mínimo al 75 % del total de sesiones.
- b) Aprobar la evaluación parcial práctica o el recuperatorio de práctica.
- c) Aprobar el Taller.

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: Presentación, prueba y defensa de la actividad del taller que deberá ser solicitado con una anticipación de 20 (veinte) días. Duración 2 hs.

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

Primera etapa: Evaluación escrita.

Temas a desarrollar, ejemplos y problemas correspondientes a todas unidades. Duración 4 hs. Modalidad escrita.

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



.....
Msc. Ing. Margarita María Álvarez