

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

**PLANIFICACIÓN ANUAL 2024**

# **Asignatura: Lenguajes de Programación y Compiladores**

**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:** 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:** 2024

**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	
<b>CARGA HORARIA TOTAL DE LA ACTIVIDAD CURRICULAR</b>	<b>75 hs.</b>

**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 t3pico que trata la asignatura es la aplicaci3n de las gram3ticas regulares y libres de contexto y, de los aut3matas finitos y de pila en los compiladores e int3rpretes. Se proporciona la justificaci3n te3rica y pr3ctica de los instrumentos necesarios para la construcci3n de los compiladores.

Los conceptos y t3cnicas correspondientes al tema Compiladores le permitir3n al alumno desarrollar un int3rprete sencillo y adem3s, pueden ser utilizadas en el desarrollo de software de aplicaci3n general.

El valor formativo de todos los conceptos desarrollados en la asignatura es esencial en el dise1o de Compiladores e Int3rpretes y adem3s, para proveer una base s3lida 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 Programaci3n y Teor3a de la Computaci3n.

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

El Licenciado en Sistemas de Informaci3n es un profesional que:

*Posee:*

- o Los conocimientos b3sicos (l3gico-matem3ticos y computacionales) para una adecuada fundamentaci3n te3rica de su quehacer profesional espec3fico.

*Est3 capacitado para:*

- o Identificar, formular y resolver problemas de Inform3tica.
- o Gestar, dise1ar y desarrollar proyectos de Inform3tica.
- o Gestionar, planificar, ejecutar y controlar proyectos de Inform3tica.
- o Utilizar t3cnicas y herramientas de aplicaci3n en la Inform3tica.
- o Comunicarse de manera efectiva en el 3mbito profesional.
- o Tiene una actitud flexible para integrar equipos interdisciplinarios en el desarrollo y administraci3n de proyectos de Sistemas Inform3ticos y de Sistemas de Informaci3n.

## 2.4- INTEGRACI3N HORIZONTAL Y VERTICAL CON OTRAS ASIGNATURAS.

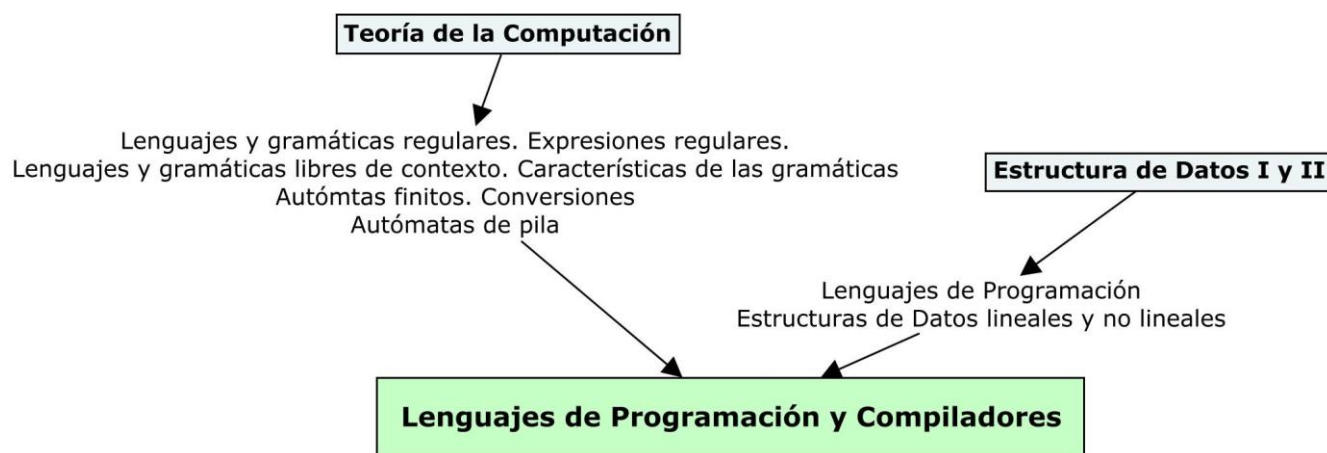


Figura 1. Integraci3n horizontal y vertical con otras asignaturas

## 3. OBJETIVOS

- a. Conocer los diferentes conceptos de los lenguajes de programaci3n.
- b. Diferenciar los criterios de dise1o y de implementaci3n de lenguajes de programaci3n para ser capaces de justificar una elecci3n.

- 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

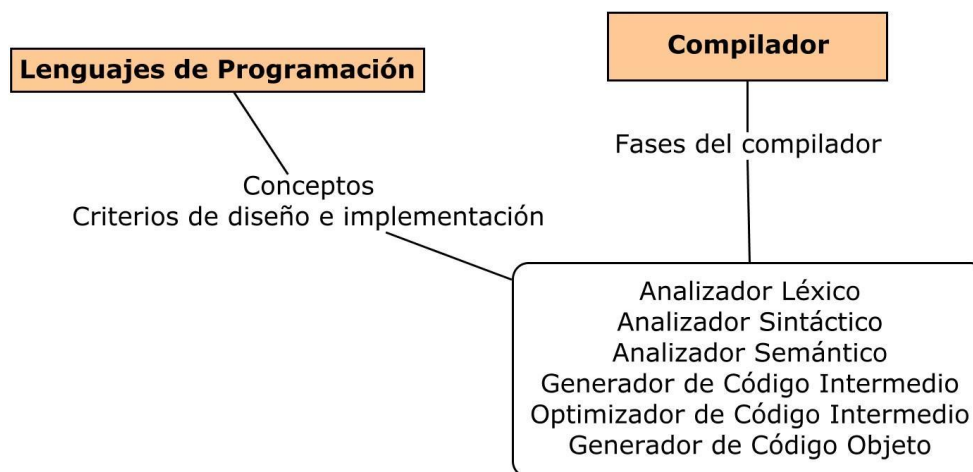


Figura 1. Articulación temática de la asignatura

### 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 actividades prácticas 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 estudiante 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 los lenguajes de programación y 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.

#### **Trabajo Práctico 1: Analizador léxico**

- 1) Identificar componentes léxicos a partir de especificaciones y gramáticas libres de contexto dadas.
- 2) Formular patrones que generen los componentes léxicos identificados.
- 3) Identificar y resolver los problemas que se presentan en el diseño y programación de un analizador léxico en Java.
- 4) Identificar y resolver los problemas que se presentan al construir un analizador léxico utilizando el generador ANTLR.
- 5) Consultar patrones a un LLM y establecer diferencias con los que realiza el estudiante.
- 6) Solicitar a un LLM programas de reconocimiento de componentes léxicos, ejecutar y verificar su correctitud.
- 7) Trabajo de investigación, el cual se comunica por escrito, para complementar el estudio de analizadores léxicos, mediante la búsqueda e interpretación de papers actuales, sobre uno de los siguientes temas: Análisis Léxico Basado en Trigramas y N-gramas, Técnicas de Parallel Scanning y Técnicas de Backtracking Controlado.

#### **Trabajo Práctico 2: Analizador Sintáctico y generación y optimización de código intermedio**

- 1) Resolver problemas del reconocimiento de programas usando los analizadores sintácticos predictivos no recursivos y los analizadores pertenecientes a la familia de analizadores LR.
- 2) Identificar y resolver los problemas que se presentan al construir un analizador sintáctico utilizando el generador ANTLR.
- 3) Resolver problemas de traducción usando los siguientes código intermedios: notación postfija, árbol sintáctico y GDA, código de 3 direcciones, triplas y cuádruplas.
- 4) Identificar y resolver problemas de optimización de los códigos intermedios usando las técnicas de subexpresión común, propagación de copias, código inactivo y optimización de lazos.
- 5) *Trabajo de investigación:* Seleccionar uno de los siguientes temas: Parsers de Redes Neuronales, Análisis Sintáctico Probabilístico, Enfoques Híbridos basados en aprendizaje automático y Verificación Formal y, realizar un trabajo de investigación para complementar el estudio de analizadores sintácticos.

#### **Trabajo Práctico 3: Conceptos de los lenguajes de programación**

- 1) Trabajo de investigación, para complementar el estudio de los conceptos de lenguajes de programación, mediante la búsqueda e interpretación de papers actuales, sobre los criterios de diseño e implementación de lenguajes de programación modernos, tales como: Python, JavaScript, Ruby, etc.
- 2) Realizar una búsqueda de los lenguajes de programación más requeridos actualmente en el mercado laboral, y analizar sus principales características en términos de sus paradigmas (imperativo, orientado a objetos, funcional, lógico, etc.), manejo de memoria, tipado, y facilidad de uso para ciertos tipos de aplicaciones.
- 3) Para cada uno de los lenguajes descriptos en el punto anterior, determinar cuál es la característica de diseño que los hace diferente a otros lenguajes en uso (de ser posible).
- 4) Seleccionar uno de los lenguajes y determinar si se le podría agregar el manejo de una nueva estructura de datos, una nueva forma de iteración, o un mecanismo de concurrencia.

## ACTIVIDADES DE RESOLUCIÓN DE PROBLEMAS DEL MUNDO REAL

### TALLER: DISEÑO Y CONSTRUCCIÓN DE UN INTÉRPRETE

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.

#### OBJETIVOS:

- (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.
- (iii) Adquirir habilidades para construir compiladores/intérpretes sencillos.

#### ACTIVIDADES

1. Revisar la documentación técnica de análisis y diseño del lenguaje y de los dispositivos para generarlo realizada en la asignatura Teoría de la Computación.
  - Verificar la completitud y correctitud de:
    - componentes léxicos y expresiones regulares,
    - gramática libre de contexto.
  - Realizar ajustes en caso necesario.
2. Diseñar y construir el Analizador Lexicográfico. Codificarlo mediante el empleo del generador automático de analizadores léxicos.
3. Diseñar y construir el Analizador Sintáctico. Codificarlo mediante el empleo del generador automático de analizadores sintácticos.
4. Diseñar y construir la fase de interpretación y, los módulos de interfaz de usuario y manejo de tabla de símbolos .
5. Probar que funcione correctamente el Analizador Lexicográfico, el Analizador Sintáctico, la fase de interpretación y los restantes módulos.
6. 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.
7. 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:** Lenguaje de programación y ANTLR

#### EVALUACIÓN

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 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. Funcionamiento del intérprete: se evaluará con los siguientes criterios:

- a) Completitud de las operaciones exigidas.
  - b) Funcionamiento correctamente.
  - c) Correcta detección de errores y emisión de los mensajes correspondientes.
  - d) Facilidad de uso.
  - e) Adecuado diseños de pantallas 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 y de vocabulario técnico.
  - c. Correcto uso de recursos gráficos y de técnicas de comunicación oral y, una adecuada organización de la información.
4. Trabajo en grupo: se evaluarán de acuerdo a los siguientes criterios de evaluación:
- a. Adecuado nivel de involucramiento de cada miembro en las tareas asignadas.
  - b. Adecuada comunicación entre los miembros del equipo.
  - c. Realización de una autoevaluación del desempeño de cada miembro del equipo.

**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
<b>Identificación, formulación y resolución de problemas de informática</b>	<p><b>Trabajo Práctico 1: Analizador léxico</b></p> <ul style="list-style-type: none"> <li>• Identificar componentes léxicos a partir de especificaciones y gramáticas libres de contexto dadas.</li> <li>• Formular patrones que generen los componentes léxicos identificados.</li> <li>• Identificar y resolver los problemas que se presentan en el diseño y programación de un analizador léxico en Java.</li> <li>• Identificar y resolver los problemas que se presentan al construir un analizador léxico utilizando el generador ANTLR.</li> </ul> <p><b>Trabajo Práctico2: Analizador Sintáctico y generación y optimización de código intermedio</b></p> <ul style="list-style-type: none"> <li>• Resolver problemas del reconocimiento de programas usando los analizadores sintácticos ascendentes y descendentes.</li> <li>• Identificar y resolver los problemas que se presentan al construir un analizador sintáctico</li> </ul>	<ul style="list-style-type: none"> <li>• Identifica las funcionalidades de los analizadores.</li> <li>• Identifica componentes léxicos y formula patrones.</li> <li>• Identifica y resuelve problemas que se presentan al desarrollar un análisis léxico en Java y con ANTLR.</li> <li>• Resuelve el problema de reconocimiento de un programa mediante analizadores sintácticos.</li> <li>• Resuelve problemas de traducción a código intermedios.</li> <li>• Identifica y resuelve problemas de optimización de los</li> </ul>	<b>Alto</b>



Eje	Actividades	Resultados de Aprendizaje	Grado de Profundidad en el tratamiento
	utilizando el generador ANTLR. <ul style="list-style-type: none"> <li>Resolver problemas de traducción usando los diferentes códigos intermedios.</li> <li>Resolver problemas de optimización de los códigos intermedios usando las técnicas de optimización.</li> </ul>	códigos intermedios.	
Las actividades planteadas en los trabajos prácticos 1, 2 y Taller: <i>Diseño y construcción de un intérprete</i> , se corresponden con la formación del eje transversal descripto ya que los estudiantes afianzan en la identificación y resolución de problemas que se presentan en el diseño y construcción de analizadores léxicos y sintácticos, en la generación y optimización de código intermedio.			

Eje	Actividades	Resultados de Aprendizaje	Grado de Profundidad en el tratamiento
<b>Concepción, diseño y desarrollo de proyectos de informática</b>	<b>Taller: <i>Diseño y construcción de un intérprete.</i></b> <ul style="list-style-type: none"> <li>Revisar la documentación técnica de análisis y diseño del lenguaje y de los dispositivos para generarlo realizada en la asignatura Teoría de la Computación.               <ul style="list-style-type: none"> <li>Verificar la completitud y correctitud de: componentes léxicos, expresiones regulares y gramática libre de contexto.</li> <li>Realizar ajustes en caso necesario.</li> </ul> </li> <li>Diseñar y construir el Analizador Lexicográfico.</li> <li>Diseñar y construir el Analizador Sintáctico.</li> <li>Diseñar y construir la fase de interpretación y, los módulos de interfaz de usuario y manejo de tabla de símbolos.</li> <li>Probar que funcione correctamente el Analizador Lexicográfico, el Analizador Sintáctico, la fase de interpretación y los restantes módulos.</li> </ul>	<ul style="list-style-type: none"> <li>Diseñe y desarrolle el intérprete siguiendo las especificaciones previstas en el análisis y diseño.</li> <li>Pruebe y corrija las funciones desarrolladas.</li> </ul>	<b>Medio</b>
Las actividades planteadas se corresponden con la formación del eje transversal descripto ya que los estudiantes realizan el desarrollo de un intérprete. Para ello, los estudiantes deben revisar la documentación del lenguaje y de los dispositivos para generarlo que fueron analizados y diseñados en asignatura previa y diseñar y desarrollar el intérprete. Por lo que, el estudiante realiza actividades que contribuyen al acervo de conocimientos y habilidades para el desarrollo de un proyecto informático.			

Eje	Actividades	Resultados de Aprendizaje	Grado de Profundidad en el tratamiento
<b>Utilización de técnicas y herramientas de aplicación en la informática</b>	<b>Trabajo Práctico 1: Analizador léxico y Trabajo Práctico 2: Analizador Sintáctico y generación y optimización de código intermedio</b> <ul style="list-style-type: none"> <li>Construir analizadores léxicos utilizando el lenguaje de programación Java y el generador ANTLR.</li> <li>Construir un analizador sintáctico utilizando el generador ANTLR.</li> </ul> <b>Taller: <i>Diseño y construcción de un intérprete.</i></b> <ul style="list-style-type: none"> <li>Diseñar y construir el Analizador Lexicográfico.</li> <li>Diseñar y construir el Analizador Sintáctico.</li> <li>Diseñar y construir la fase de interpretación y, los módulos de interfaz de usuario y manejo de tabla de símbolos.</li> </ul>	<ul style="list-style-type: none"> <li>Instale y use la herramienta de generación de analizadores: ANTLR.</li> <li>Instale y use el lenguaje de programación Java.</li> <li>Aplica las técnicas y herramientas para crear los analizadores.</li> <li>Programa los módulos para el manejador de la tabla de símbolos, interfaz de usuario y para la interpretación.</li> </ul>	<b>Medio</b>

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: habilidad para identificar y utilizar los recursos tecnológicos necesarios que den soporte al desarrollo del intérprete y, capacidad para desenvolverse en entornos de desarrollo de software libre o propietario.

Eje	Actividades	Resultados de Aprendizaje	Grado de Profundidad en el tratamiento
<b>Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.</b>	<b>Taller:</b> Diseño y construcción de un intérprete (todas las actividades descriptas)	Diseña y desarrolla un intérprete sobre temas/ lenguajes actuales.	<b>Alto</b>

El intérprete, producto de las actividades realizadas en el Taller: *Diseño y construcción de un intérprete*, se considera un desarrollo tecnológico ya que los intérprete y compiladores, permiten el desarrollo de software de manera más efectiva, optimizan los recursos computacionales y, facilitan la creación de nuevas tecnologías y aplicaciones.

Eje	Actividades	Resultados de Aprendizaje	Grado de Profundidad en el tratamiento
<b>Fundamentos para el desempeño en equipos de trabajo</b>	<b>Taller:</b> Diseño y construcción de un intérprete <ul style="list-style-type: none"> <li>Trabajar en equipos. Para ello:               <ul style="list-style-type: none"> <li>Crear el equipo de trabajo.</li> <li>Realizar los acuerdos iniciales de trabajo (formas y frecuencias de las reuniones).</li> <li>Planificar las actividades.</li> <li>Asignar roles, responsabilidades y tareas a cada integrante.</li> </ul> </li> <li>Evaluar el desempeño de los integrantes del equipo de trabajo.</li> </ul>	<ul style="list-style-type: none"> <li>Organice el grupo de trabajo para realizar las tareas de creación del grupo.</li> <li>Distribuya responsabilidades y actividades a cada integrante del grupo de trabajo.</li> <li>Evalué el desempeño de los integrantes y del trabajo en equipo.</li> </ul>	<b>Medio</b>

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
<b>Fundamentos para la comunicación efectiva</b>	<b>Taller:</b> Diseño y construcción de un intérprete. <ul style="list-style-type: none"> <li>Elaborar el informe técnico del diseño y construcción de todas las fases del intérprete.</li> <li>Exponer el trabajo realizado en el taller.</li> </ul> <b>Trabajo Práctico 1: Analizador léxico</b> <ul style="list-style-type: none"> <li>Elaborar un informe producto de la investigación sobre el tema: nuevas técnicas utilizadas en el analizador léxico.</li> </ul> <b>Trabajo Práctico 2: Analizador Sintáctico y generación y optimización de código intermedio</b> <ul style="list-style-type: none"> <li>Elaborar un informe producto de la investigación sobre el tema: nuevas tecnologías de analizadores sintácticos.</li> </ul> <b>Trabajo Práctico 3: Conceptos de los lenguajes de programación</b> <ul style="list-style-type: none"> <li>Elaborar un informe producto de la investigación sobre el tema: criterios de diseño e implementación de lenguajes de programación modernos.</li> </ul>	<ul style="list-style-type: none"> <li>Redacta un informe técnico que documente de manera clara, estructurada y coherente el proceso de construcción de un intérprete y de las actividades de investigación realizadas, asegurando que toda la información relevante esté bien organizada y sea fácilmente comprensible.</li> <li>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.</li> <li>Presenta de manera clara y coherente los resultados del trabajo realizado en el taller,</li> </ul>	<b>Medio</b>

		utilizando un lenguaje técnico adecuado. • Integra en su presentación la prueba del prototipo para apoyar a la comprensión.	
<p>Las actividades de redacción y exposición del trabajo realizado en el taller: Diseño y construcción de un intérprete y las actividades de investigación descritas en los trabajos prácticos que requieren un informe por escrito, se corresponde con la formación de este eje transversal 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.</p> <p>Además, los estudiantes deben exponer el trabajo realizado en el taller: Diseño y construcción de un intérprete afianzando la competencia de comunicación oral mediante la utilización correcta de técnicas de comunicación oral y de vocabulario técnico.</p>			

Eje	Actividades	Resultados de Aprendizaje	Grado de Profundidad en el tratamiento
<b>Fundamentos para el aprendizaje continuo</b>	<p><b>Trabajo Práctico 1: Analizador léxico</b></p> <ul style="list-style-type: none"> <li>Trabajo de investigación, para complementar el estudio de analizadores léxicos, mediante la búsqueda e interpretación de papers actuales, sobre uno de los siguientes temas: Análisis Léxico Basado en Trigramas y N-gramas, Técnicas de Parallel Scanning y Técnicas de Backtracking Controlado.</li> </ul> <p><b>Trabajo Práctico 2: Analizador Sintáctico y generación y optimización de código intermedio</b></p> <ul style="list-style-type: none"> <li>Seleccionar uno de los siguientes temas: Parsers de Redes Neuronales, Análisis Sintáctico Probabilístico, Enfoques Híbridos basados en aprendizaje automático y Verificación Formal y, realizar un trabajo de investigación para complementar el estudio de analizadores sintácticos.</li> </ul> <p><b>Trabajo Práctico 3: Conceptos de los lenguajes de programación</b></p> <ul style="list-style-type: none"> <li>Trabajo de investigación, para complementar el estudio de los conceptos de lenguajes de programación, mediante la búsqueda e interpretación de papers actuales, sobre los criterios de diseño e implementación de lenguajes de programación modernos, tales como: Python, JavaScript, Ruby, etc.</li> </ul>	<ul style="list-style-type: none"> <li>Investiga y evalúa información de fuentes web actualizadas y confiables sobre los temas especificados relacionados con técnicas que se usan actualmente en el desarrollo de analizadores léxicos y sintácticos y sobre los criterios de diseño e implementación de lenguajes de programación modernos.</li> <li>Desarrolla habilidades de aprendizaje autónomo, buscando, comprendiendo y contextualizando información nueva sobre los lenguajes de programación y compiladores.</li> </ul>	<b>Bajo</b>

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.

### 5.3 CRONOGRAMA DE FORMACIÓN PRÁCTICA

ACTIVIDAD	CARGA HORARIA	CRONOGRAMA DE DESARROLLO
<b>Trabajo Práctico 1 – Analizador Léxico e Introducción a ANTLR</b>	9 hs	1º, 2º y 3º Semana (se prevén 2 horas del total para el desarrollo del AL del taller)
<b>Trabajo Práctico 2 (parte I) – Analizador Sintáctico</b>	9 hs	4º, 5º y 6º Semana (se prevén 2 horas del total para el desarrollo del AS del taller)

ACTIVIDAD	CARGA HORARIA	CRONOGRAMA DE DESARROLLO
<b>Trabajo Práctico 2 (parte II) – Generación y Optimización de código</b>	9 hs	7°, 8° y 9° Semana (se prevén 2 horas del total para el desarrollo del A Semántico del taller)
<b>Taller: Diseño y construcción de un intérprete.</b>	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.
<b>Parcial de Práctica</b>	3 hs	13° semana
<b>Recuperatorio Parcial</b>	3 hs	14° semana
<b>Presentación del Taller</b>	3 hs	13° semana
<b>Defensa del Taller</b>	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. 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 las funcionalidades de los analizadores.	Identifica correctamente las funcionalidades clave de los analizadores, tales como análisis léxico, sintáctico y semántico.
Identifica componentes léxicos y formula patrones.	Identifica correctamente los componentes léxicos relevantes dentro de un texto o lenguaje específico, como palabras clave, identificadores, operadores, literales, y delimitadores.
Identifica y resuelva problemas que se presentan al desarrollar un análisis léxico en Java y con ANTLR.	Identifica correctamente los problemas típicos que pueden surgir durante el desarrollo de un análisis léxico con ANTLR, como la ambigüedad en las reglas léxicas, conflictos de precedencia, o manejo incorrecto de tokens.
Resuelva el problema de reconocimiento de un programa mediante analizadores sintácticos.	Precisión en la selección de los algoritmos de análisis sintáctico de acuerdo con las características de las gramáticas. Correcta aplicación de los algoritmos de análisis sintáctico.
Resuelva problemas de traducción a código intermedios.	Utiliza correctamente las técnicas de traducción a código intermedio
Identifica y resuelva problemas de optimización de los códigos intermedios.	Aplica correctamente las técnicas de optimización de código intermedio.
Diseña y desarrolle el intérprete siguiendo las especificaciones previstas en el análisis y diseño.	Diseña y desarrolla un intérprete que cumple con todas las especificaciones previstas en la metodología dada. El diseño es claro y bien documentado. El desarrollo del intérprete demuestra un conocimiento profundo de los conceptos de interpretación.
Pruebe y corrija las funciones desarrolladas.	Realiza pruebas exhaustivas y detalladas de todas las funciones desarrolladas, identificando y corrigiendo errores. Utiliza casos de prueba variados y representativos para asegurar la robustez y la funcionalidad correcta de las funciones requeridas. El programa funciona correctamente.
Instale y use la herramienta de generación de analizadores: ANTLR.	Instala el programa requerido resolviendo las situaciones excepcionales que se le pudieran presentar durante la instalación. Demuestra dominio de la herramienta de generación de analizadores.
Instale y use el lenguaje de programación Java.	Instala el programa requerido resolviendo las situaciones excepcionales que se le pudieran presentar durante la instalación. Demuestra dominio de la herramienta de programación.
Aplique las técnicas y herramientas para crear los analizadores.	Correcta selección de técnicas para implementar analizadores. Correcta selección de herramientas para implementar analizadores.
Programe los módulos para el manejador de la tabla de símbolos, interfaz de usuario y para la interpretación.	Cumple adecuadamente todos los requerimientos de programación.
Interpreta los errores.	Capacidad para interpretar errores y explicar una posible alternativa de solución.

### 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
<b>Evaluación 1:</b> Introducción a la Construcción de Compiladores y Analizador Léxico	3º semana
<b>Evaluación 2:</b> Analizador Sintáctico	6º semana
<b>Evaluación 3:</b> Conceptos de lenguajes, Analizador Semántico y Traducción dirigida por la sintaxis	10º semana
<b>Evaluación 4:</b> Generación de Código intermedio, Optimización de Código Intermedio, Generación de código Objeto y tabla de símbolos.	13º semana

## CRONOGRAMA DE EVALUACIONES DE PRÁCTICA

Evaluación	Fecha
<b>Evaluación Parcial:</b> Los analizadores sintácticos. Generación y optimización de código	8º semana
<b>Evaluación Recuperatoria:</b> Los analizadores sintácticos. Generación y optimización de código	10º semana
<b>Defensa del Taller:</b> exposición oral y ejemplificación del funcionamiento de los analizadores.	13º y 14º semana

### 8.3.2- CRITERIOS DE EVALUACIÓN

Los criterios de evaluación de los *parciales teóricos* son:

El estudiante debe ser capaz de:

**Evaluación 1:** Introducción a la Construcción de Compiladores y Analizador Léxico Reconocer y comprender las fases de un compilador:

1. Comprender correctamente la estructura de un compilador y las funciones de cada fase.
2. Interpretar correctamente el significado de lexemas, componentes léxicos y patrones.
3. Definir claramente la especificación y reconocimiento de componentes léxicos.

**Evaluación 2:** Analizador Sintáctico

1. Conocer correctamente la clasificación de los analizadores sintácticos.
2. Interpretar adecuadamente los algoritmos de análisis sintáctico.

**Evaluación 3:** Conceptos de lenguajes, Analizador Semántico y Traducción dirigida por la sintaxis

1. Identificar claramente los conceptos constructivos de los lenguajes de programación.
2. Identificar correctamente las funciones del analizador semántico y comprender los formalismos de traducción dirigida por la sintaxis.

**Evaluación 4:** Generación de Código intermedio, Optimización de Código Intermedio, Generación de código Objeto, tabla de símbolos y criterios de diseño e interpretación de los lenguajes de programación.

1. Conocer correctamente las distintas representaciones intermedias y reconocer las técnicas de optimización de código.
2. Identificar correctamente las funciones del generador de código objeto.
3. Conocer adecuadamente las estructuras usadas en la tabla de símbolo.
4. Explicar con precisión los criterios de diseño e interpretación de los lenguajes de programación.

Los criterios de evaluación a aplicar en la *evaluación parcial* y en el *recuperatorio de práctica* son los que se detallan a continuación:

El estudiante debe ser capaz de:

- Interpretar y aplicar correctamente los algoritmos de análisis sintáctico.
- Traducir correctamente un código a una representación intermedia.
- Realizar la optimización de código en forma correcta.

Los criterios de evaluación del Taller: *Diseño y Construcción de un Intérprete* están consignados en la descripción del mismo (apartado 5.1). 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 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 80 % del total de sesiones.
- b) Aprobar la evaluación práctica parcial con un puntaje  $\geq 70$  puntos
- c) Aprobar las evaluaciones teóricas con un puntaje  $\geq 70$  puntos
- d) Aprobar el Taller con un puntaje  $\geq 70$  puntos.

#### 8.5.2- CONDICIONES PARA LOGRAR LA REGULARIDAD DE LA ASIGNATURA

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