



**INSTITUTO/S:** Tecnología e Ingeniería

**CARRERA/S:** Licenciatura en Informática

**MATERIA:** Parseo y generación de código

**NOMBRE DEL RESPONSABLE DE LA ASIGNATURA:** Pablo Miguel Angel Pandolfo

**EQUIPO DOCENTE:** -----

**CUATRIMESTRE:** 1<sup>ro</sup>

**AÑO:** 5<sup>to</sup>

**PROGRAMA N°:** 43

**(Aprob. Por Cons.Directivo fecha 24/08/2022)**

**Instituto/s:** Tecnología e Ingeniería

**Carrera/s:** Licenciatura en Informática.

**Nombre de la materia:** Parseo y generación de código

**Responsable de la asignatura y equipo docente:** Pablo Miguel Angel Pandolfo

**Cuatrimestre y año:** 1<sup>er</sup> cuatrimestre del 5<sup>to</sup> año.

**Carga horaria semanal:** 4 horas

**Programa N°:** 43

**Código de la materia en SIU:** 784

## **Parseo y generación de código**

### **1. Fundamentación**

Un compilador es un programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación, generando un programa equivalente que la máquina será capaz de interpretar. Usualmente el segundo lenguaje es lenguaje de máquina, pero también puede ser un código intermedio, o simplemente texto. Este proceso de traducción se conoce como compilación.

La asignatura presenta al alumno las nociones fundamentales para la construcción de un compilador y brinda las herramientas necesarias para su construcción.

### **2. Propósitos y/u objetivos**

#### **Objetivos**

Que el/la estudiante:

- Entienda la arquitectura de herramientas existentes de análisis y síntesis de programas tales como intérpretes, compiladores y analizadores de código.
- Sea capaz de concebir soluciones a nuevos problemas de análisis y síntesis de programas.
- Desarrolle herramientas de análisis y síntesis de programas de pequeña escala para afianzar los conocimientos adquiridos.

### **3. Programa sintético:**

Estructura de compiladores. Compilación vs. interpretación. Estructura de un compilador. Análisis léxico. Análisis sintáctico. Métodos ascendentes y descendentes. Tabla de símbolos. Árboles de parsing y árboles de sintaxis abstracta. Tratamiento de errores. Análisis semántico.

Verificación de tipos. Generación de código. Optimización de código.

#### **4. Programa analítico**

##### **4.1 Organización del contenido:**

###### **Unidad 1: Introducción.**

Definición de compilador. Estructura de un compilador. Fases del proceso de compilación. Herramientas y descripción del lenguaje.

###### **Unidad 2: Análisis Léxico.**

Utilidad del análisis léxico. Funcionamiento. Términos utilizados. Especificación del analizador léxico. Construcción de un analizador léxico.

###### **Unidad 3: Análisis Sintáctico.**

Funciones del analizador sintáctico. Diseño de gramáticas. Dificultades para la creación de gramáticas. Análisis sintáctico lineal. Diagramas de sintaxis. ASD. Analizadores sintácticos predictivos. Conjuntos de predicción y gramáticas LL(1). Conversión a gramáticas LL(1). Analizadores sintácticos descendentes recursivos. Implementación de ASP's. ASA. Algoritmo de desplazamiento y reducción. Construcción de tablas de análisis sintáctico SLR. Organigrama de las gramáticas.

###### **Unidad 4: Análisis Semántico.**

Introducción. Tabla de tipos. Tabla de símbolos. Introducción. Atributos y acciones semánticas. Tipos de atributos. Definición dirigida por la sintaxis (DDS). Esquema de traducción (ETDS). Comprobaciones semánticas.

###### **Unidad 5: Generación de código.**

Introducción. Tipos de código intermedio. Código intermedio para expresiones, asignaciones, sentencias de entrada y salida, sentencia condicional. Iteraciones tipo while, do-while, for. Selección. Código intermedio para vectores, registros, subprogramas no recursivos y recursivos.

##### **4.2 Bibliografía y recursos obligatorios:**

**Aho A.; Sethi, R.; Ullman, J.D.** (1990). Compiladores: Principios, técnicas y herramientas, 1ra. edición, México: Addison-Wesley.

**Ruiz Catalán, J** (2010). Compiladores. Teoría e Implementación, 1ra. edición, España, Madrid: Alfaomega.

##### **4.3 Bibliografía optativa:**

No aplica.

#### **5. Metodología de enseñanza:**

Las clases serán en modalidades teórica, práctica y teórico-práctico dependiendo del tema a desarrollar.

En las clases teóricas se reforzará con un material de lectura (artículos, capítulos de libros y links a sitios de internet) y en algunos casos con cuestionarios para realizar a través del campus.

Se dará a los/as alumnos/as guías de trabajos prácticos con casos prácticos ejercicios para realizar con el objetivo de asentar los conceptos trabajados en clase. Algunos de estos ejercicios formarán parte de su evaluación.

### **Plan de trabajo en el campus:**

El Campus Virtual es un espacio fundamental para el desarrollo de la asignatura. En el aula virtual se propondrá material educativo, apuntes de clase, bibliografía, así como también el programa y cronograma de la asignatura y las guías de trabajos prácticos.

## **6. Actividades de investigación y extensión (si hubiera).**

En la materia no se desarrollan actividades de investigación en el marco de un proyecto de investigación. No obstante, algunos puntos de las guías de trabajos prácticos se presentan algunos puntos que requieren que los/as alumnos/as investiguen para poder resolverlos.

## **7. Evaluación y régimen de aprobación**

### **7.1 Aprobación de la cursada**

Para aprobar la cursada y obtener la condición de regular, el régimen académico establece que debe obtenerse una nota no inferior a cuatro (4) puntos. Todas las instancias evaluativas deberán tener una instancia de recuperatorio. Podrán acceder a la administración de esta modalidad solo aquellos y aquellas estudiantes que hayan obtenido una nota inferior o igual a 6 (seis) puntos en el examen parcial.

Siempre que se realice una evaluación de carácter recuperatorio, la calificación que los/as estudiantes obtengan reemplazará la calificación obtenida en el examen que se ha recuperado y será la considerada definitiva a los efectos de la aprobación.

El/La alumno/a deberá poseer una asistencia no inferior al 75% en las clases presenciales.

En cuanto a la cursada virtual se requerirá que el/la estudiante ingrese al aula virtual como mínimo una vez por semana.

### **7.2 Aprobación de la materia**

La materia puede aprobarse por promoción, evaluación integradora, examen final o libre.

**Promoción directa:** tal como lo establece el art°17 del Régimen Académico, para acceder a esta modalidad, el/la estudiante deberá aprobar la cursada de la materia con una nota no inferior a siete (7) puntos, no obteniendo en ninguna de las instancias de evaluación parcial menos de seis (6) puntos, sean evaluaciones parciales o recuperatorios. El promedio estricto resultante deberá ser una nota igual o superior a siete (7) sin mediar ningún redondeo.

**Evaluación integradora:** tal como lo establece el art°18 del Régimen Académico, podrán acceder a esta evaluación aquellos estudiantes que hayan aprobado la cursada con una nota de entre cuatro (4) y seis (6) puntos.

La evaluación integradora tendrá lugar por única vez en el primer llamado a exámenes finales posterior al término de la cursada. Deberá tener lugar en el mismo día y horario de la cursada y será administrado, preferentemente, por el/la docente a cargo de la comisión. Se aprobará tal instancia con una nota igual o superior a cuatro (4) puntos, significando la aprobación de la materia.

La nota obtenida se promediará con la nota de la cursada.

**Examen final:** Instancia destinada a quienes opten por no rendir la evaluación integradora o hayan regularizado la materia en cuatrimestres anteriores. Se evalúa la totalidad de los contenidos del programa de la materia y se aprueba con una calificación igual o superior a cuatro (4) puntos. Esta nota no se promedia con la cursada.

### 7.3 Criterios de calificación

El sistema normal de evaluación consistirá en 2 (dos) exámenes parciales con recuperatorios, según el cronograma previsto, de la totalidad de la materia descrita en el programa. Los mismos se realizarán en las fechas que, a tal efecto, se establezcan en el cronograma. El primer parcial es teórico-práctico e individual y el segundo parcial es un trabajo integrador desarrollado por los/as alumnos/as de manera grupal y se debe defender por todos los integrantes del grupo. Además, se considera como parte de la evaluación de la cursada el desarrollo de la guía de trabajos prácticos que los/as alumnos/as deben presentar de manera grupal al final de la cursada de la asignatura.

## 8. Cronograma

CLASE	UNIDAD	TEMAS	ACTIVIDADES	MODALIDAD
1	Introducción	Presentación de la materia. Conceptos fundamentales.	Preguntas teóricas. Armar mapa conceptual. Definición del lenguaje para el cual se va a diseñar y desarrollar un compilador.	Virtual. Sincrónica
2	Introducción	Estructura de un compilador.	Ejercicios de partes de un compilador. Investigar herramientas de diseño y desarrollo de compiladores.	Virtual. Sincrónica
3	Análisis Léxico	Utilidad, funcionamiento y especificación del Scanner.	Crear diagramas de transiciones (DT), tabla de transiciones y programa en java para implementar la tabla. Definir las especificaciones léxicas del lenguaje del TP.	Virtual Sincrónica

4	Análisis Sintáctico	Funciones del Parser. Diseño de gramáticas. Diagrama de Sintaxis.	Diseñar gramáticas no ambiguas para diferentes lenguajes. Crear diagramas de sintaxis y programas en java para las gramáticas. Definir las especificaciones sintácticas del lenguaje del TP.	Virtual Sincrónica
5	Análisis Sintáctico	ASD con retroceso y predictivos LL(1)	Comprobar si una gramática es LL(1). Si lo es, construir tabla de parsing para determinar si una entrada es aceptada o no.	Virtual Sincrónica
6	Análisis Sintáctico	ASA con retroceso y predictivo SLR.	Construir autómatas a partir de las colecciones canónicas y la tabla de análisis SLR.	Virtual Sincrónica
7	Analisis Sintactico	Repaso General	Ejercicios varios	Virtual Sincrónica
8	<b>Examen Parcial</b>			Virtual Sincrónica
9	Análisis Sintáctico		Resolución del parcial.	Virtual Sincrónica
10	Análisis Semántico	Tabla de Tipos	Crear la tabla de tipos y símbolos para un programa en un lenguaje dado.	Virtual Sincrónica
11	Análisis Semántico	Tabla de Símbolos	Crear la tabla de tipos y símbolos para un programa en un lenguaje dado.	Virtual Sincrónica
12	Análisis Semántico	Atributos y acciones semánticas. Comprobaciones semánticas. DDS y ETDS.	Construir DDS y ETDS  Definir las especificaciones semánticas del lenguaje del TP.	Virtual Sincrónica
13	Generación de código	Expresiones, asignaciones, sentencias de entrada y salida y condicional. iteraciones tipo while, do-while, for.	Ejercicios de generación de código	Virtual Sincrónica
14	Generación de código	Vectores, registros y subprogramas no recursivos y recursivos.	Ejercicios de generación de código	Virtual Sincrónica
15	<b>Presentación TP</b>		Implementación de un compilador.	Virtual Sincrónica

16	Recuperatorios		Finalización del cuatrimestre. Indicaciones para la evaluación integradora.	Virtual Sincrónica
----	----------------	--	--	-----------------------