

Carrera INGENIERIA EN INFORMATICA		
Asignatura 3663 - Lenguajes y Compiladores		
Trayecto Infraestructura		
Año académico 2023		
Responsable / Jefe de cátedra Ing. Hernan Villarreal		
Carga horaria semanal 4hs	Carga horaria total 64hs	Créditos -----
Modalidad: Presencial		
Correlativas anteriores AUTOMATAS Y GRAMATICAS	Correlativas posteriores ELECTIVA I	
Conocimientos necesarios -----		

<p>Descripción de la asignatura</p> <p>La asignatura Lenguajes y Compiladores aporta conocimientos vitales para la asimilación del funcionamiento de los lenguajes de programación y su construcción en general. También plantea la adquisición de conocimientos básicos de compiladores, su funcionamiento, implementación y su importancia en el proceso de construcción de software.</p>
<p>Metodología de enseñanza</p> <p>Se dedica el 40% del tiempo a clases teórico/prácticas expositivas, 20% a la resolución de ejercicios prácticos y el resto a realizar la construcción de un compilador. Este trabajo práctico único se lo enfoca como un caso de estudio y la casi totalidad del tiempo se dedica al desarrollo en máquina.</p> <p>Las clases teóricas son dictadas de manera presencial y además poseen un soporte digitalizado. Las mismas se encuentran en diapositivas (Microsoft PowerPoint) animadas con audio accesibles desde la plataforma MiEL.</p> <p>Es así como la visión de compiladores se torna eminentemente pragmática, prestándose especial atención a las técnicas y estrategias constructivas. Se tratan esencialmente los temas de análisis lexicográfico, análisis sintáctico, generación de código y versatilidad de las plataformas. Por último, se incluyen aspectos avanzados de optimización de código y de tratamiento de errores.</p>
<p>Objetivos de aprendizaje</p> <ul style="list-style-type: none"> • Lenguajes de programación: Una vez aprobado el curso el alumno estará en condiciones de identificar los principales aspectos sintácticos y semánticos y de identificar las principales características de un lenguaje de programación para asociarlas con el modelo teórico de ejecución del mismo.

Objetivos específicos: Estará en condiciones de identificar las principales limitaciones de un lenguaje de programación y mitigar las mismas con un adecuado diseño de la aplicación.

- **Compiladores:** Una vez aprobado el curso el alumno estará en condiciones de construir un compilador básico, para un lenguaje de programación.
Objetivos específicos: Estará en condiciones de construir un analizador léxico y un analizador sintáctico tanto con el uso de herramientas o sin ellas. Estará en condiciones de construir un generador de código para cualquier plataforma. Estará en condiciones de seleccionar representaciones intermedias y de agregarles optimizaciones de código específicas para cada una de ellas.
- **Objetivos comunes a Lenguajes / Compiladores:** Una vez aprobado el curso el alumno estará en condiciones de reconocer problemas en los que es posible diseñar un lenguaje para su solución y diagramar una construcción de un compilador para dicho lenguaje.
- Este conocimiento aporta una gran versatilidad para afrontar el aprendizaje futuro de nuevos lenguajes de programación acompañando al avance tecnológico.

Contenidos mínimos

Autómatas finitos y expresiones regulares. Propiedades. Gramáticas libres del contexto. Propiedades. Lenguajes deterministas. Gramáticas LR. Fases de un compilador. Análisis lexicográfico. Análisis Sintáctico. Generadores de analizadores sintácticos. Análisis ascendente y descendente. Traducción dirigida por sintaxis. Generación de código intermedio y final: su optimización.

Competencias a desarrollar

Genéricas

- Utilización de técnicas y herramientas de aplicación en la ingeniería en sistemas de información / informática.
- Desempeño en equipos de trabajo.
- Comunicación efectiva.
- Actuación profesional ética y responsable.
- Aprendizaje continuo.
- Desarrollo de una actitud profesional emprendedora.
- Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.
- Identificación, formulación y resolución de problemas de ingeniería en sistemas de información/informática.

Específicas

- Especificación, proyecto y desarrollo de software.
- Procedimientos y certificaciones del funcionamiento, condición de uso o estado de sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software.

- Dirección y control de la implementación, operación y mantenimiento de sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software.

Programa analítico	
Unidad 1	Perspectiva histórica. Primeros lenguajes de alto nivel. Relaciones entre los diferentes lenguajes de programación. Diagrama de interrelación entre los diferentes lenguajes, basado en la influencia de unos en otros.
Unidad 2	Estructura de los lenguajes de programación. Sintaxis, semántica y pragmática. Procesamiento de los lenguajes. Interpretación. Compilación. Lenguajes estáticos. Lenguajes basados en la pila. Lenguajes dinámicos.
Unidad 3	Sintaxis. Reglas léxicas. Reglas gramaticales. Metalenguajes. BNF. Árbol de Parsing. Impacto de la recursividad sobre la evaluación de expresiones. Influencia de la sintaxis sobre la claridad: formato de las instrucciones, comentarios, puntuación.
Unidad 4	Introducción a los compiladores. Compilación e interpretación. Ensambladores y compiladores. Cantidad de pasadas. Conversores fuente-fuente. Compiladores incrementales. Autocompiladores. Metacompiladores. Compilaciones Separadas. Vinculación. Descompiladores. Salidas del compilador. Notaciones para representar implementaciones. Diagramas T.
Unidad 5	Estructura general de un compilador. Relación entre los tipos de gramáticas y la estructura de los compiladores. Tabla de símbolos. Tabla de palabras reservadas. Mecanismo de acceso.
Unidad 6	Analizador léxico. Construcción de autómatas finitos a partir de gramáticas regulares. Implementación de analizadores léxicos mediante estructuras de control clásicas y mediante arreglos de punteros. Utilización de Flex.
Unidad 7	Tabla de símbolos. Problemas de alcance. Relación entre la estructura en tiempo de ejecución y la tabla de símbolos. Etapas en la construcción de la tabla de símbolos.
Unidad 8	Análisis sintáctico. Análisis sintáctico descendente. Análisis sintáctico ascendente. Retrocesos. Detección y reporte de errores. Análisis sintáctico descendente predictivo. Conflictos desplazamiento-reducción. Conflictos reducción-reducción. Utilización de un analizador sintáctico. Bison/Yacc.
Unidad 9	Árboles sintácticos. Traducción guiada por la sintaxis. Semántica estática. Comprobación de tipos. Grafo de las alternativas en la generación de código: árbol sintáctico, tercetos, tercetos indirectos, cuartetos, notación polaca inversa. Caminos posibles.
Unidad 10	Generación de código. Generación de código intermedio. Generación de código ejecutable. Relleno con retroceso. Nociones de manejo de errores. Nociones de optimización de código.
Unidad 11	Tipos de Datos. Su tratamiento en compiladores.

	El concepto de binding. Entidades y atributos. Binding dinámico y binding estático. Variables. Binding de alcance, de almacenamiento, de valor y de tipo. Binding de tipo e interpretación. Unidades de programa. Estructura en tiempo de ejecución.
Unidad 12	El concepto de binding. Entidades y atributos. Binding dinámico y binding estático. Variables. Binding de alcance, de almacenamiento, de valor y de tipo. Binding de tipo e interpretación. Unidades de programa. Estructura en tiempo de ejecución.

Planificación de actividades					
Semana	Clase	Actividad	Tipo	Duración estimada	Unidad
Semana 1	Lenguajes de programación Analizador Léxico y Expresiones regulares	Clase oral	Teoría. Practica aplicada		1 / 2
Semana 2	Analizador Léxico y Expresiones regulares. Vinculación. Compiladores e interpretes. Metacompiladores.	Clase oral	Teoría. Practica aplicada		/ 3 / 4 / 5 / 6
Semana 3	Análisis sintáctico. Tabla de símbolos	Clase oral	Teoría. Practica aplicada		7 / 8
Semana 4	Parsing. Árboles sintácticos.	Clase oral	Teoría. Practica aplicada		9
Semana 5	Consultas generales de compilación en etapas de AL / AS	Clase oral	Teoría. Practica aplicada		
Semana 6	Parcial I	Examen			
Semana 7	Generación de código intermedio. Sentencias Básicas.	Clase oral	Teoría. Practica aplicada		10 / 11
Semana 8	Generación de código intermedio. Sentencias de Control.	Clase oral	Teoría. Practica aplicada		10/ 11
Semana 9	Generación de Código Assembler. Coprocesador Matemático.	Clase oral	Teoría. Practica aplicada		10 / 11
Semana 10	Tipos de Datos. Optimización de Código.	Clase oral	Teoría. Practica aplicada		11

Semana 11	Binding. Estructura en tiempo de ejecución.	Clase oral	Teoría. Practica aplicada		12
Semana 12	Parcial II	Examen			
Semana 13	Consultas generales de compilación en etapas de GCI / GC / Tipos	Clase oral	Teoría. Practica aplicada		
Semana 14	Recuperatorio	Examen			
Semana 15	Cierre	Clase oral			

Evaluación

Se evaluará a través de dos exámenes parciales.

Habrá un solo examen recuperatorio para el primer o segundo examen parcial. Los exámenes serán escritos y presenciales, conteniendo preguntas y ejercicios que permitan enunciar, describir, aplicar y relacionar algunos de los conceptos presentados a largo del curso como así también explicar, analizar, resolver y relacionar los conceptos aprendidos con el trabajo practico enunciado (compilador).

Los contenidos a evaluar en el 1er Parcial son:

Autómatas finitos y expresiones regulares. Gramáticas libres del contexto. Propiedades. Lenguajes deterministas. Gramáticas LR. Análisis lexicográfico. Análisis Sintáctico. Generadores de analizadores sintácticos. Análisis ascendente y descendente.

Los contenidos a evaluar en el 2do Parcial son:

Traducción dirigida por sintaxis.
Generación de código intermedio y final: su optimización.
Fases de un compilador.

En el día de la entrega de notas a los alumnos, se realiza la resolución del examen frente al curso, lo que habilita la evacuación de dudas y consultas referidas a los ejercicios tomados.

Promoción:

Se obtiene aprobando los parciales o su recuperatorio con 7 (siete) o más puntos.

Examen libre:

Con dos meses de anticipación a la fecha de sustanciación de mesa, el alumno deberá entregar la resolución de todas las actividades prácticas además del trabajo práctico "compilador" de forma completa.
Para obtener el tema especial para el trabajo práctico deberá comunicarse con los docentes en forma personal.
El examen libre abarcará los contenidos de todas las unidades temáticas. El alumno será evaluado en primer lugar con un examen escrito.
De estar aprobado este examen deberá rendir un examen de forma oral.

Trabajo práctico:

Cada grupo o alumno, recibe consignas específicas para su compilador que hace que sea diferente de los restantes. Estas consignas personalizadas contienen elementos léxicos, elementos sintácticos y elementos semánticos. Durante este trabajo el alumno debe entonces realizar una actividad importante fuera del horario de clase para lo que cuenta con una plataforma en la que puede exponer sus consultas y un foro de debate, moderado y supervisado por los docentes.

Primera evaluación	Semana 6	Evaluación presencial escrita	2.5 hs. , 19 hs.
Segunda evaluación	Semana 12	Evaluación presencial escrita	2.5 hs. , 19 hs.
Recuperatorio	Semana 15	Evaluación presencial escrita	2.5 hs. , 19 hs.

Bibliografía obligatoria				
Título	Autor	Editorial	Edición	Año
Modern Compiler Implementation in Java	Appel A. W.	Cambirdge University Press	2da	1997
Programming Language Concepts	Ghezzi Carlo	John Wiley & Sons	3ra	1997
Compiladores, Principios, Técnicas y Herramientas	Aho A. V., Lam, M., Sethi, R. Ullman J.	Pearson Addison-Wesley Iberoamericana	1ra	2008

Otros recursos obligatorios (videos, enlaces, otros) Incluir una fila por cada recurso	
Nombre	

Otros recursos complementarios (videos, enlaces, otros) Incluir una fila por cada recurso	
Nombre	