

UNIVERSIDAD NACIONAL DEL SUR BAHÍA BLANCA		1 5
DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN		
COMPILADORES E INTERPRETES	CÓDIGO: 5576	
	ÁREA N°: II	

CARRERAS Licenciatura en Ciencias de la Computación					
PROFESOR RESPONSABLE: Dra. María Laura Cobo – Profesora Adjunta con Dedicación Exclusiva Dr. Sebastián Gottifredi – Profesor Adjunto con Dedicación Simple					
CARGA HORARIA	Teoría 64hs	Práctica 64 hs	Laboratorio	CANTIDAD DE SEMANAS	16
CORRELATIVAS					
PARA CURSAR LA MATERIA		PARA APROBAR LA MATERIA			
APROBADAS	CURSADAS Lenguajes de Programación	APROBADAS Lenguajes de Programación	CURSADAS		
DESCRIPCIÓN					
Es un curso de diseño de compiladores e intérpretes en el que se estudian los conceptos fundamentales de la traducción de lenguajes de programación y se analizan distintas formas de resolver los problemas que se presentan al diseñar el traductor de un lenguaje, independientemente de las máquinas fuente y destino que se elijan. El curso agrupa, combina y pone en práctica varios conceptos de ciencias de la computación vistos en la carrera. A nivel de teoría de computación conceptos como expresiones regulares y gramáticas libres de contexto; a nivel de sistemas conceptos vinculados a lenguajes de programación, sistemas operativos, organización de memoria; y a nivel de ingeniería de software conceptos de versiones, testing y patrones de diseño. A lo largo del curso los alumnos desarrollan un proyecto de mediana complejidad que consiste en un compilador para un subconjunto de algún lenguaje (que incluya unidades, recursividad, tipos de datos simples y alguno estructurado). La traducción se hace a código intermedio. Como lenguaje de implementación, los alumnos pueden optar entre Java y C++. Si bien es poco probable que un alumno llegue a programar un compilador para un lenguaje de programación, las técnicas presentadas en la materia y la implementación de algunas de ellas generan una comprensión más acabada del funcionamiento de las facilidades del mismo permitiendo una mejor elección de este tipo de herramientas en el futuro. También les permite mapear los conocimientos de las técnicas desarrolladas para compiladores a otras áreas de interés, principalmente del área de desarrollo de software. Con este curso se espera que los alumnos, además de construir conocimiento relacionado con los contenidos conceptuales que se describen en el programa, desarrollen competencias para: • Desarrollar un proyecto de mediana complejidad, • Trabajar en forma colaborativa, • Investigar, • Mejorar la expresión oral y escrita,					

UNIVERSIDAD NACIONAL DEL SUR BAHÍA BLANCA		2 5
DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN		
COMPILADORES E INTERPRETES	CÓDIGO: 5576	
	ÁREA N°: II	

- Redactar manuales en forma correcta,
- Confrontar diferentes posturas sobre un mismo tema,
- Argumentar sobre sus propias opiniones y decisiones.
- Aprovechar conocimientos y/o técnicas adquiridos en otras áreas

METODOLOGÍA DE ENSEÑANZA

En las clases teóricas la metodología de enseñanza es **expositiva**, colectiva y con un enfoque **deductivo**. Cada contenido conceptual se presenta ilustrando su motivación y algorítmica, se presentan ejemplos concretos en cada caso para lograr la correcta comprensión de los diferentes conceptos, métodos y herramientas. Si bien se promueve la participación de los alumnos el profesor conduce la clase de acuerdo al cronograma establecido. Los contenidos son presentados al grupo completo de alumnos y el profesor detiene la explicación cuando surgen dudas individuales. En general los ejercicios de fijación de conceptos se resuelven en forma colaborativa con los alumnos.

En las clases prácticas se adopta un enfoque más **participativo e individual**. Para cada parte del proyecto se realizan clases **expositivas y colectivas** donde se enfatizan los requerimientos del mismo como así también las posibles herramientas que se requieran para el desarrollo.

Por último, cabe destacar que en las clases teóricas el proyecto tiene un rol fundamental ya que es utilizado como recurso didáctico para ejemplificar cómo los nuevos conceptos se aplican evolutivamente sobre desarrollos previos, y para contrastar soluciones alternativas a problemas ya resueltos. De esta manera, la cátedra promueve una sinergia constante entre las actividades teóricas y las actividades prácticas realizadas por los alumnos.

MECANISMO DE EVALUACIÓN

El cursado de la asignatura es evaluado mediante un proyecto realizado en el transcurso de todo el cuatrimestre. Este proyecto se realiza en etapas que son evaluadas en forma independiente. La cátedra asigna a cada comisión de alumnos un docente auxiliar para evaluar el proyecto. La última etapa del proyecto consiste en una defensa oral de los desarrollos realizados en las etapas anteriores. Al finalizar el cuatrimestre se toma como nota de coloquio la nota de evaluación del proyecto realizado. De esta manera para aprobar la materia restar rendir un segundo coloquio, oral, en el que se evalúan los conceptos teóricos no evaluados durante el cursado de la materia.

En cada etapa del proyecto se evaluarán:

- los aspectos prácticos y teóricos de los temas de la materia respecto a la correctitud de la resolución de las tareas requeridas por la etapa.
- la calidad de los informes presentados
- la calidad de la batería de tests desarrollada.

En las instancias de evaluación oral se buscará detectar el nivel de comprensión de los temas teóricos trabajados y la correcta integración de los nuevos conceptos con los anteriormente estudiados.

UNIVERSIDAD NACIONAL DEL SUR BAHÍA BLANCA		3 5
DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN		
COMPILADORES E INTERPRETES	CÓDIGO: 5576	
	ÁREA N°: II	

Además, estas evaluaciones buscan mejorar la capacidad de los alumnos de expresar oralmente ideas, defender posturas y responder a desafíos.

PROGRAMA SINTÉTICO

- Compiladores, ensamblados e intérpretes. Traductores.
- Relación con lenguajes formales y teoría de autómatas.
- Estructura de un compilador.
- Fases de un compilador.
- Un compilador sencillo de una pasada.
- Traducción dirigida por la sintaxis.
- Análisis léxico. Función y construcción de un analizador léxico. Generadores de analizadores léxicos.
- Análisis sintáctico. Función y construcción del analizador sintáctico. Generadores de analizadores sintácticos.
- Análisis semántico. Función y construcción del analizador semántico.
- Sistemas de tipos. Comprobador de tipos.
- Ambientes para el momento de la ejecución.
- Generación de código intermedio.
- Generación de código.
- Optimización de código.
- Intérpretes. Estructura y construcción.
- Validación de un traductor.

PROGRAMA ANALÍTICO

- 1) Traductores: Compiladores, ensambladores e intérpretes. Evolución histórica. Contexto de un compilador. Lenguajes involucrados en el proceso de traducción de un lenguaje de programación. Número de pasadas de un compilador. Relación con lenguajes formales y teoría de autómatas. Aplicaciones.
- 2) Estructura de un compilador. Especificación de un lenguaje. Gramáticas. Expresiones regulares. Autómatas. Definiciones dirigidas por la sintaxis. Diseño arquitectónico de un compilador. Fases de un compilador. Análisis y síntesis en el proceso de traducción. Agrupamiento de las fases. Necesidad de la tabla de símbolos. Detección e información de errores.
- 3) Análisis léxico. Scanning. Función del analizador léxico. Recursos de especificación. Formas de implementación del analizador léxico y del scanner. Optimización. Interacción con la Tabla de Símbolos. Errores léxicos. Generadores de analizadores léxicos. Aplicaciones.
- 4) Análisis sintáctico. Función del analizador sintáctico. Analizadores sintácticos basados en gramáticas libres de contexto. El problema de la ambigüedad. Métodos ascendentes y métodos descendentes. Análisis LL(1). Eliminación de la recursión a izquierda y factorización.

DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

COMPILADORES E INTERPRETES

CÓDIGO: 5576

ÁREA N°: II

Construcción de analizadores LL(1) basados en tabla y desdentes recursivos. Análisis LR(0) y LR(1): métodos SLR, LALR y LR canónico. Tratamiento de errores sintácticos para cada uno de los métodos presentados. Recuperación de errores. Uso de generadores de analizadores sintácticos. Aplicaciones.

5) Análisis semántico. Vinculación con el analizador sintáctico y el número de pasadas. Definiciones dirigidas por la sintaxis y esquemas de traducción. Atributos. Definiciones con atributos por la izquierda. Evaluación de reglas semánticas. Acciones semánticas. Evaluación de atributos sintetizados y heredados en un análisis ascendente y en uno descendente.

6) Estructuras y representaciones intermedias para el análisis semántico. Tablas de símbolos y árboles sintácticos abstractos. Estructura, entradas y nodos. Patrones de diseño vinculados. Construcción y vinculación con el analizador sintáctico. Funcionalidades, recorridos y controles. Interacción entre las estructuras.

7) Controles semánticos específicos. Controles de declaraciones y especificaciones. Resolución de nombres. Chequeo estático de tipo. Conversiones de tipo. Sobrecarga de funciones y operadores. Funciones polimórficas. Interacción con la tabla de símbolos y los árboles sintácticos abstractos. Manejo de errores: detección y recuperación.

8) Generación de código intermedio. Roles de las representaciones intermedias en un traductor. Lenguajes intermedios. Uso de definiciones dirigidas por la sintaxis para la generación del código intermedio. Generación a partir de un árbol sintáctico abstracto. Generación a partir de una representación de bajo nivel. Código intermedio para una máquina virtual y código intermedio para un generador de código.

9) Generación de código. Aspectos generales. La máquina objeto. Administración de la memoria en tiempo de ejecución. Uso de los registros rápidos de la máquina destino. Construcción de un generador de código simple.

10) Optimización de código. Significado. Costo. Principales fuentes para la optimización. Tipos de optimización. Soluciones para diferentes situaciones optimizables.

11) Otros problemas de traducción. Interpretes. Estructura de un intérprete. Tipos de Interpretes. El rol de la Tabla de Símbolos en un intérprete. Construcción de un intérprete. Compilación Just-in-time. Transpilación. Web browsers. Herramientas formateo de documentos y de representación gráfica.

12) Verificación y validación de un compilador. Conceptos. Principios para la construcción de los tests de prueba. Desarrollo de una batería de tests. Testeo de regresión en un traductor. Proceso de validación.

BIBLIOGRAFÍA

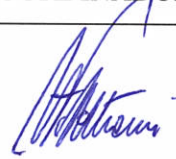
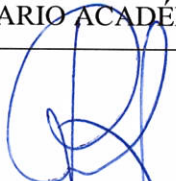
UNIVERSIDAD NACIONAL DEL SUR BAHÍA BLANCA		5
DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN		5
COMPILADORES E INTERPRETES	CÓDIGO: 5576	
	ÁREA N°: II	

Bibliografía Básica

- Scott, Michael L.; *Programming Languages Pragmatics*. 4ta Edición. Morgan Kaufmann Publishers. 2015.
- Cooper, Keith D.; Torczon, Linda: "*Engineering a Compiler*" 2ª ed. Morgan Kaufmann in an imprint of Elsevier. 2012.
- Grune, Dick; van Reeuwijk, Kees; Bal, Henri E.; Jacobs, Criel J. H.; Langendoen, Koen *Modern Compiler Design*. Second Edition. Springer. 2012.
- Manuales desarrollados por la cátedra:
 - de sintaxis del lenguaje base a implementar
 - de semántica del lenguaje base a implementar
 - del lenguaje de implementación
 - de la máquina virtual CeIVM

Bibliografía Adicional

- Grune, Dick; Jacobs, Criel J. H. *Parsing Techiques: A Practical Guide* Second Edition. Springer. 2008.
- Aho, A. V.; Lam, M. S.; Sethi, R.; Ullman, J. D.: "*Compilers. Principles, Techniques and Tools*" 2ª ed. Addison-Wesley. 2007.
- Apple, A. *Modern compiler implementation in C*. Cambridge University Press. 2008.
- Apple, A. *Modern compiler implementation in Java*. Cambridge University Press. 2002.
- Kakde, O. G.: *Algorithms for Compiler Design*. Charles River Media. 2002.
- Kowaltowski. - *Implementacao de Linguagens de Programacao*.
- Louden, K. *Compiler construction. Principles and practice*. PWS Publishing Company. 1997.
- Mogensen T.A. *Basics of Compiler Desing*. 2010.
- Watt, D.; Brown, D.: *Programming Language Processors in Java: Compilers and Interpreters*. Prentice Hall. 2000.

AÑO		FIRMA PROFESOR RESPONSABLE	
2018		 	
VISADO			
COORDINADOR ÁREA	SECRETARIO ACADÉMICO	DIRECTOR DEPARTAMENTO	
	 Dr. DIEGO C. MARTINEZ SECRETARIO ACADÉMICO DPTO. DE CS. E ING. DE LA COMP. UNIVERSIDAD NACIONAL DEL SUR	 Dr. MARCELO A. FALAPPA DIRECTOR DECANO DPTO. DE CS. E ING. DE LA COMP. UNIVERSIDAD NACIONAL DEL SUR	