

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS

**NOMBRE DEL CURSO: Organización de Lenguajes y Compiladores 2**

CODIGO:	781	CREDITOS:	5
ESCUELA:	Ciencias y Sistemas	AREA A LA QUE PERTENECE:	Ciencias de la Computación
PRE REQUISITO:	777 - Organización de Lenguajes y Compiladores 1 772 - Estructuras de datos	POST REQUISITO:	281 - Sistemas Operativos 1
CATEGORIA:	Obligatorio	SEMESTRE:	Primer Semestre 2020
EDIFICIO:	T1, T7, T3	SECCIÓN:	A, B+,B-, C
HORAS POR SEMANA DEL CURSO:	4 horas	HORAS POR SEMANA DEL LABORATORIO:	2 horas
DÍAS QUE SE IMPARTE EL CURSO SECCION A:	Lunes y sábado	HORARIO DEL CURSO SECCION A:	7:10 AM – 8:50 AM 12:10 PM – 13:50 PM
DÍAS QUE SE IMPARTE EL CURSO SECCION B+, B-:	Lunes y viernes	HORARIO DEL CURSO SECCION B+,B-:	7:10 AM – 8:50 AM
DÍAS QUE SE IMPARTE EL CURSO SECCIÓN C	Miercoles	HORARIO DEL CURSO SECCION C	9:10 AM – 10:30 AM
CATEDRÁTICOS:		TUTORES ACADEMICOS:	
Sección A:	Ing. Byron López	Sección A:	Luis Lizama
Sección B+:	Ing. Edgar Sabán	Sección B+:	Pavel Vásquez
Sección B-	Ing. Erick Navarro	Sección B-:	Rainman Sián
Sección C	Ing. Luis Espino	Sección C:	Juan Carlos Maeda

DESCRIPCIÓN DEL LABORATORIO

En este laboratorio el estudiante podrá poner en práctica los conceptos que se desarrollen en la clase magistral, como la traducción dirigida por la sintaxis, generación y optimización de código intermedio; que son útiles en muchos otros contextos más allá de compiladores, como ingeniería de software y seguridad.

Quizás el resultado más útil del curso es que los estudiantes comprendan profundamente las capacidades y limitaciones de los compiladores modernos y cómo pueden ser utilizados

eficazmente. Este conocimiento no solo es importante para aspirantes a diseñadores de lenguajes, sino también para depurar y optimizar casi cualquier aplicación.

COMPETENCIAS ESPECÍFICAS (CE) DE LAS ACCIONES FORMATIVAS DE LA DISCIPLINA

1. Reconoce, entiende y es capaz de aplicar los conceptos de lenguajes formales y de programación, como la definición de lenguajes formales, gramáticas y su clasificación, análisis léxico y sintáctico descendente (predictivo recursivo).
2. Interpreta, analiza y aplica conceptos y procedimientos de Organización de Lenguajes y Compiladores 1 para la solución de problemas de análisis ascendente, traducción e interpretación de lenguajes definidos por gramáticas libres de contexto.
3. Utiliza software para la generación de analizadores léxicos y sintácticos.
4. Razona crítica y lógicamente sobre los procesos y resultados para verificar su validez por medio de la comparación con el conocimiento y la experiencia.

METODOLOGÍA

- Se desarrollarán ejercicios prácticos, que serán la implementación de los ejercicios resueltos en la clase magistral, se mostrará cómo realizar un front-end para el desarrollo de un compilador utilizando como base el apéndice A del libro de texto.
- Se realizarán evaluaciones presenciales prácticas en los periodos de laboratorio, lo cual permitirá evaluar el aprendizaje y la aplicación de los conceptos adquiridos.
- Además de la bibliografía recomendada, se proporcionará al alumno material complementario que le ayude a conocer distintas técnicas en la elaboración de compiladores.
- Se realizarán dos proyectos para poder evaluar los conceptos adquiridos en clase, tomando en cuenta que pueden incluirse temas de cursos pre-requisito.

EVALUACIÓN DEL RENDIMIENTO ACADÉMICO

El laboratorio tiene una ponderación de 34 puntos distribuidos de la siguiente manera:

Actividad	Ponderación	Porcentaje
Primer proyecto	15.3	45%
Segundo proyecto	18.7	55%
Total	34	100%

Observaciones:

- Para aprobar el laboratorio se debe tener una nota final igual o mayor al 61% de los puntos, es decir 20.74 puntos de 34.
- Copias parciales o totales de los proyectos tendrán una nota de 0 puntos y los responsables serán reportados a la Escuela de Ingeniería en Ciencias y Sistemas.
- Se deben enviar los archivos entregables en las fechas establecidas y por el medio indicado para tener derecho a calificación.

SESIONES

Las clases de laboratorio estarán divididas en sesiones, siendo una por semana y la misma para los laboratorios de todas las secciones. Estas sesiones se dividirán en 3 tipos de sesiones según el tipo de contenido a abarcar, estas son:

1. **Sesiones de laboratorio:** Estas serán las sesiones en las que se cubrirá el contenido del laboratorio.
2. **Sesiones de resolución de parcial:** Estas sesiones serán 3 únicamente, como su nombre lo indica se estará dando la resolución del parcial y serán dadas en la misma semana que se realizó el parcial.

CONTENIDO

Para su mayor comprensión el contenido del laboratorio se dividirá en 5 unidades,

Unidad 1: Repaso de cursos anteriores

- Procesadores de lenguaje
- La estructura de un compilador
 - Fases de un compilador
 - Fase de análisis
 - Fase de síntesis
 - Administración de la tabla de símbolos
- Mecanismos para el paso de parámetros
- Llamada por valor
- Llamada por referencia
- Llamada por nombre
- Definición de sintaxis

❖ **Sesiones estimadas: 1 – 2 Sesiones de laboratorio**

Unidad 2: Traducción dirigida por la sintaxis

- Esquemas de traducción orientados por la sintaxis
 - Con acciones dentro de las producciones
 - Eliminación de la recursividad por la izquierda de esquemas de traducción
 - Con atributos heredados por la izquierda
- Aplicaciones de la traducción dirigida por la sintaxis
 - Expresiones aritméticas con un esquema de traducción sencillo
 - Construcción de árboles de análisis sintáctico
 - Expresiones
 - Recorrido de un árbol de análisis sintáctico (interprete)
 - Obtener el valor implícito de una expresión
 - Validación de errores
 - Comprobación de tipos
 - Conversión de tipos
- Construcción de árboles de análisis sintáctico

- Sentencias de selección
- Sentencias cíclicas
- Sentencias de transferencia
- Recorrido de un árbol de análisis sintáctico
 - Validación de errores
 - Comprobación de tipos
 - Estructuras y técnicas auxiliares para manejo de sentencias de transferencia
- Construcción de árboles de análisis sintáctico
 - Métodos y funciones
- Recorrido de un árbol de análisis sintáctico
 - Validaciones semánticas para métodos y funciones
 - Instanciación
 - Acceso a atributos y métodos
- Construcción de árboles de análisis sintáctico
 - Arreglos
 - Clases y objetos
- Recorrido de un árbol de análisis sintáctico
 - Acceso a arreglos
 - Validación de errores en arreglos
 - Definición consistente

❖ **Sesiones estimadas: 3 sesiones de laboratorio, 1 sesión de resolución de parcial**

Unidad 3: Introducción a la generación de código intermedio

- Modelo básico orientado a objetos para la generación de código intermedio
 - ¿Qué es lo mínimo que se necesita para generar código intermedio?
 - Árbol de análisis sintáctico
 - Manejar los punteros principales
 - Administración de la tabla de símbolos para un generador de código intermedio
 - Elementos esenciales de un símbolo
 - Tablas de símbolos encadenadas por alcance
 - ◆ La regla del bloque anidado más cercano
 - ◆ Definición de entorno
 - ◆ Implementación de un entorno
- Construcción y recorrido de árbol para generación de código de tres direcciones
 - Traducción de expresiones
 - Traducción de sentencias de selección
 - Traducción de sentencias cíclicas
 - Traducción de sentencias de transferencia
 - Sentencia break
 - Sentencia continue
 - Salto incondicional goto
 - **Display**

❖ **Sesiones estimadas: 2 sesiones de laboratorio y 1 sesión de resolución de parcial**

Unidad 4: Optimización de código intermedio

- Fundamentos de optimización de código intermedio
 - Definición, objetivos e importancia de la optimización de código intermedio
- Optimización por bloques
- Optimización por mirilla

❖ **Sesiones estimadas: 2 sesiones de laboratorio, 1 sesión de resolución de parcial**

CALENDARIZACION DE ACTIVIDADES

Actividad	Publicación de enunciado	Fecha de entrega	Días
Primer proyecto	Miércoles 12 de febrero	Domingo 29 de marzo	46
Segundo proyecto	Lunes 30 de marzo	Viernes 15 de mayo	47

ENTREGA DE PROYECTOS

- La entrega de cada uno de los proyectos es individual.
- No se recibirán proyectos después de la fecha de entrega.

CALIFICACIÓN DE PROYECTOS

- La calificación de los proyectos se realizará presencialmente y desde los archivos ejecutables.
- No se puede agregar o quitar algún símbolo en el archivo de entrada. El proyecto deberá funcionar con los archivos que sean proveídos por lo auxiliares para la calificación, sin modificación.
- No será permitido compartir los archivos de entrada durante ni después de la calificación.
- La calificación del proyecto será personal y existirá un tiempo límite. Se debe tomar en cuenta que no pueden estar personas ajenas a la calificación, de lo contrario no se calificará el proyecto.
- Anomalías o copias detectadas de proyectos tendrán de manera automática una nota de 0 puntos y los involucrados serán reportados a la Escuela de Ingeniería en Ciencias y Sistemas, para que se apliquen las sanciones correspondientes.
- Existirán horarios para la calificación de cada proyecto, por el cual el estudiante deberá de elegir el horario que mejor le convenga.
- Anomalías detectadas en los archivos entregables tendrá de manera automática una nota de 0 puntos, por ejemplo: no se envió el código correcto, se envió parte del código y no el código completo, archivos ajenos a los entregables del proyecto, no se hizo uso de las herramientas descritas en el enunciado de cada proyecto, entre otras.

BIBLIOGRAFÍA

JLex <https://www.cs.princeton.edu/~appel/modern/java/JLex/current/manual.html>

Cup <http://www2.cs.tum.edu/projects/cup/docs.php>

Dos Reis, A. (2012) Compiler Construction Using Java, JavaCC and Yacc.
JohnWiley & Sons, Inc. Publication

Appel, A; Palsberg, J. (2004) Modern Compiler Implementation in Java.
Cambridge University Press, 2ed.