

NOMBRE DEL CURSO: Organización de Lenguajes y Compiladores 2

CÓDIGO:	781	CRÉDITOS:	5
ESCUELA:	Ciencias y Sistemas	ÁREA 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
CATEGORÍA:	Obligatorio	SEMESTRE:	2do. Semestre 2018
EDIFICIO:	T-3	SECCIÓN:	A, B+, B-
HORAS POR SEMANA DEL CURSO:	4	HORAS POR SEMANA DEL LABORATORIO:	2
DÍAS QUE SE IMPARTE EL CURSO SECCIÓN A:	Lunes y Sábado	HORARIO DEL CURSO SECCIÓN A:	7:10 AM – 8:50 AM 12:30 PM – 14:10 PM
DÍAS QUE SE IMPARTE EL CURSO SECCIÓN B+ y B-:	Lunes y Viernes	HORARIO DEL CURSO SECCIÓN B+ y B-:	7:10 AM – 8:50 AM

DESCRIPCIÓN DEL LABORATORIO:

El laboratorio del curso de Organización de Lenguajes y Compiladores 2, trata sobre la parte práctica del curso. Se encarga de la aplicación de la segunda y última fase del compilador, la fase de síntesis. Se refuerza los conceptos de la fase de análisis y se hace interactuar el análisis léxico con el sintáctico y semántico a través de definiciones dirigidas por la sintaxis. Por último, se realiza la generación de código intermedio y se optimiza dicho código con las técnicas indicadas en el contenido. Para realizar los proyectos del laboratorio se utilizarán distintas herramientas de creación de analizadores léxico y sintáctico.

OBJETIVO GENERAL:

Poner en práctica los conceptos aprendidos sobre el desarrollo de un compilador.

OBJETIVOS ESPECÍFICOS:

1. Que el estudiante sea capaz de desarrollar aplicaciones que sean capaces de leer una entrada y producir una acción de salida.

2. Que el estudiante entienda el funcionamiento de las herramientas que utiliza diariamente para desarrollar diferentes aplicaciones de software.
3. Fomentar en el estudiante el análisis en la resolución de problemas complejos en la lectura de entradas determinadas para un compilador.
4. Que el estudiante tenga el conocimiento suficiente para identificar el lenguaje que mejor se adapte a los requerimientos de lo que se desea hacer.

HABILIDADES:

1. Comprensión de la estructura y funciones de un compilador.
2. Comprensión de las estructuras de datos utilizadas en el proceso de compilación y su relación con las distintas fases del compilador.
3. Comprensión de la importancia y aplicación del código intermedio en la construcción de software.
4. Comprensión de la relación entre hardware y software durante la ejecución de un programa computacional.
5. Comprensión de los requerimientos que se les planteen en los enunciados a lo largo del curso.

COMPETENCIAS:

1. Dominar los conceptos básicos de compiladores.
2. Elegir con criterio la manera óptima de codificar una solución informática.
3. Aplicar herramientas de análisis léxico y sintáctico para la resolución de problemas.
4. Crear soluciones funcionales aplicando los conceptos de compiladores.

METODOLOGÍA:

- Se impartirán clases presenciales para complementar el contenido del curso, y que el estudiante mejore sus habilidades en el diseño y construcción de compiladores.
- Actividades prácticas en los periodos de laboratorio.
- 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 prácticas y 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 36 puntos distribuidos de la siguiente manera.

Actividad	Ponderación	Porcentaje
Primer Proyecto	14	38.89%
Segundo Proyecto	22	61.11%
Total	36	100.00%

Para aprobar el laboratorio se debe tener una nota final igual o mayor al 61% de los puntos.

OBSERVACIONES:

- La calificación de los proyectos de laboratorio es personal acoplándose al día y horario que se indique previamente.
- 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 debe de mandar los archivos entregables en fechas establecidas para tener derecho a calificación.

CONTENIDO:

PRIMERA UNIDAD: Fundamentos de análisis léxico y sintáctico

Sesión 1 – Semana del 16 al 29 de julio, Clase Teórica

- 1.1. Introducción al curso y repaso.
 - 1.1.1. Definición de compilador
 - 1.1.2. Fases del compilador
 - 1.1.3. Compilador, Intérprete, traductor
- 1.2. Conceptos de análisis léxico y sintáctico
Definición, objetivos y funciones del análisis sintáctico.

2. SEGUNDA UNIDAD: Traducción orientada por la sintaxis

Sesión 2 – Semana del 30 de julio al 05 de agosto, Clase Práctica

- 2.1. Esquemas de traducción y definiciones dirigidas por la sintaxis.
 - 2.1.1. Árboles de análisis sintáctico y derivaciones
 - 2.1.2. Atributos sintetizados
 - 2.1.3. Ambigüedad
 - 2.1.4. Precedencia y asociatividad

Sesión 3 – Semana del 06 al 12 de agosto, Clase Práctica

- 2.1.5. Atributos heredados
 - 2.1.6. Análisis semántico
 - 2.1.7. Árbol de sintaxis abstracta, AST
- 2.2. Errores en el análisis sintáctico
 - 2.2.1. Estrategias de recuperación

Sesión 4 – Semana del 13 al 19 de agosto, Clase Práctica

- 2.3. Gramáticas descendentes
 - 2.3.1. Recorrido del AST
 - 2.3.2. Transformaciones
 - 2.3.3. Atributos heredados en gramáticas descendentes.

3. TERCERA UNIDAD: Generación de código intermedio

Sesión 5 – Semana del 20 al 26 de agosto, Clase Teórica-Práctica

- 3.1. Código de tres direcciones
 - 3.1.1. Definición del código de tres direcciones
- 3.2. Traducción de expresiones
 - 3.2.1. Expresiones aritméticas
 - 3.2.2. Expresiones condicionales (Corto circuito)

Sesión 6 – Semana del 27 de agosto al 02 de septiembre, Clase Práctica

- 3.3. Traducción de sentencias de flujo de control
 - 3.3.1. Display
 - 3.3.2. Códigos de salto incondicional

Sesión 7 – Semana del 03 al 09 de septiembre, Clase Teórica

- 3.4. Ambientes locales
 - 3.4.1. Pila
 - 3.4.2. Tabla de símbolos

Sesión 8 – Semana del 10 al 23 de septiembre, Clase Teórica

- 3.5. Arreglos
 - 3.5.1. Recorridos por fila
 - 3.5.2. Recorridos por columna

Semana del 24 al 30 de septiembre, semana de congresos estudiantiles.

Sesión 9 – Semana del 01 al 07 de octubre, Clase Práctica

- 3.6. Procedimientos y funciones
 - 3.6.1. Variables globales y locales.
 - 3.6.2. Parámetros por valor
 - 3.6.3. Parámetros por referencia
 - 3.6.4. Retorno de valores

4. CUARTA UNIDAD: Representación de objetos en código intermedio

Sesión 10 – Semana del 08 al 14 de octubre, Clase Teórica-Práctica

- 4.1. Manejo de cadenas
 - 4.1.1. Definición, objetivos y funciones de heap
 - 4.1.2. Referencias a heap
 - 4.1.3. Manejo de cadenas de caracteres a bajo nivel
 - 4.1.4. Operaciones sobre cadenas

Sesión 11 – Semana del 15 al 21 de octubre, Clase Teórica-Práctica

- 4.2. Manejo de objetos
 - 4.2.1. Instancias de un objeto
 - 4.2.2. Herencia
 - 4.2.3. Encapsulamiento
 - 4.2.4. Referencias a atributos

Sesión 12 – Semana del 22 al 28 de octubre, clase Teórica-Práctica

- 4.2.5. Inicialización de objetos
- 4.2.6. Métodos de objetos
- 4.2.7. Método constructor

5. QUINTA UNIDAD: Optimización de código intermedio

Sesión 13 – Semana del 29 de octubre al 03 de noviembre, Clase Teórica-Práctica

- 5.1. Fundamentos de optimización de código intermedio
 - 5.1.1. Definición, objetivos e importancia de la optimización de código intermedio
- 5.2. Optimización por bloques
 - 5.2.1. Bloques básicos
 - 5.2.2. Grafos de flujo
- 5.3. Optimización por mirilla
 - 5.3.1. Simplificación algebraica
 - 5.3.2. Eliminación de código inalcanzable
 - 5.3.3. Optimización de flujo de control

CALENDARIZACIÓN DE ACTIVIDADES:

1. Clases Prácticas:

Las clases prácticas se llevarán a cabo de dos formas:

- 1.1. **Ejercicios propuestos por los auxiliares del curso, los cuales serán resueltos en el laboratorio de forma manual.**
- 1.2. **Ejercicios propuestos por los auxiliares del curso, los cuales serán resueltos haciendo uso de alguna herramienta, entre las cuales estarán: Irony, JFlex y Cup.**

2. Primer Proyecto:

- 2.1. **Publicación de enunciado: martes 7 de agosto**
- 2.2. **Entrega: martes 18 de septiembre**

3. Segundo Proyecto:

- 3.1. **Publicación de enunciado: martes 18 de septiembre**
- 3.2. **Entrega: viernes 09 de noviembre**

4. Conferencia:

- 3.1. **Realización de conferencia sección A: segunda semana de agosto**
- 3.2. **Realización de conferencia sección B+: tercera semana de agosto**
- 3.3. **Realización de conferencia sección B-: cuarta semana de agosto**

Las conferencias a ser impartidas, son una actividad en la cual participan todas las secciones del curso.

La calendarización de las conferencias puede variar según la disponibilidad de los conferencistas.

4. Congresos estudiantiles:

- 4.1 **Semana del 24 al 30 de septiembre, semana de congresos estudiantiles.**

5. Entrega de proyectos:

- 5.1. Se publicarán links donde deberán de subir los estudiantes su proyecto, el contenido a subir deberá ser los entregables mencionados en el enunciado de cada proyecto.
- 5.2. La entrega de cada uno de los proyectos es individual.
- 5.3. Deberán ser desarrollados en los lenguajes descritos en el enunciado de cada proyecto.
- 5.4. Las herramientas a utilizar como analizadores deberán ser las mismas descritas en el enunciado de cada proyecto.
- 5.5. No se recibirán proyectos después de la fecha de entrega.

6. Calificación de proyectos:

- 6.1. La calificación de los proyectos se realizará presencialmente y desde los archivos ejecutables.
- 6.2. 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.
- 6.3. No será permitido compartir los archivos de entrada durante ni después de la calificación.
- 6.4. 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.
- 6.5. 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.
- 6.6. Existirá horarios para la calificación de cada proyecto, por el cual el estudiante deberá de elegir el horario que mejor le convenga.
- 6.7. Anomalías detectadas en los archivos entregables tendrá de manera automática una nota de 0 puntos, por ejemplo: no se envió código 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:

1. Principios, Técnicas y Herramientas Aho, Sethi y Ullman. PEARSON ADDISON- WESLEY, 2008, segunda edición.