



COMPILADORES

Sección 1



A. Información del profesor

Nombre del profesor

Ing. Max Alejandro Antonio Cerna Flores

Horario:

Lunes	19:30-21:00 Teoría
Miércoles	19:30-20:15 Teoría
Miércoles	20:15-21:00 Práctica

e-mail

macerna@correo.url.edu.gt



B. Información general

Descripción

Este curso estudia los principios básicos de un compilador y / o intérprete, partiendo de la estructura interna del proceso de compilación, y describiendo las fases de este proceso. Se tratan en detalle: análisis lexicográfico, análisis sintáctico, traducción dirigida por la sintaxis, análisis semántico y la generación del código. Para poner en práctica los conceptos aprendidos se realizan varias tareas y proyectos prácticos.

Modalidad

Presencial



C. COMPETENCIAS GENÉRICAS



El egresado landivariano se identifica por:

Pensamiento
lógico, reflexivo y
analógico

Pensamiento
crítico

Resolución de
problemas

Habilidades de
investigación

Uso de TIC y
gestión de la
información

Comunicación
efectiva, escrita y
oral

Compren
sión
lectora

Compromiso
ético y
ciudadanía

Liderazgo
constructivo

Aprecio y respeto
por la diversidad e
interculturalidad

Creatividad

COMPETENCIAS ESPECÍFICAS (propias del curso)

Competencia 1: Domina la base teórica fundamental para el entendimiento de la estructura interna del proceso de compilación.

Competencia 2: Comprende y utiliza el análisis de léxico para la definición, reconocimiento y selección de lexemas mediante herramientas de generación de autómatas y regex.

Competencia 3: Reconoce las diferentes gramáticas que existen en lenguajes regulares y sus posibles implementaciones algorítmicas en analizadores sintácticos.

Competencia 4: Implementa verificaciones de análisis semántico y type checkers en compiladores robustos y con tolerancia a errores

Competencia 5: Implementa un programa compilador hacia instrucciones de máquina virtual/ASM/tres direcciones como resultado de un árbol AST



METODOLOGÍA

Este curso se desarrollará a través de los siguientes métodos de aprendizaje-enseñanza:

Aprendizaje basado en proyectos

«Es una técnica didáctica en la que la exposición de contenido se hace por medio de Videos que pueden ser consultados en línea de manera libre, mientras el tiempo de aula se dedica a la discusión, resolución de problemas y Actividades prácticas bajo la supervisión y asesoría del profesor.» Principales tendencias pedagógicas 2016, KAREM SCHMITZ.

Aprendizaje Colaborativo

El aprendizaje colaborativo es una técnica didáctica que promueve el aprendizaje centrado en el estudiante basando el trabajo en pequeños grupos, donde las personas con diferentes niveles de habilidad utilizan una variedad de actividades de aprendizaje para mejorar su entendimiento sobre una materia.

PROGRAMACIÓN

COMPETENCIA 1

Domina la base teórica fundamental para el entendimiento de la estructura interna del proceso de compilación.

Saber conceptual (contenido temático)

- 1.1 Introducción a la compilación
- 1.2 Traductores de lenguajes de computación
- 1.3 Interpretación
- 1.4 Compilación
- 1.5 Estructura interna del proceso de compilación
- 1.6 Conceptos básicos de un compilador
- 1.7 Pasos del proceso de compilación

Saber procedimental (habilidades y destrezas)

- Establece las diferencias de las fases del proceso de compilación
- Reconoce la importancia del proceso de compilación

Saber actitudinal (conductas observables)

- Realiza paralelos entre los programas creados en el medio académico y los compiladores de la industria
- Reconoce la importancia de resolver los problemas de forma modular

Indicadores de logro 1 (resultado): Distingue las diferentes partes de un compilador clásico



COMPETENCIA 2

Comprende y utiliza el análisis de léxico para la definición, reconocimiento y selección de lexemas mediante herramientas de generación de autómatas y regex.

Saber conceptual (contenido temático)

- 2.1. Especificación de los tokens o componentes léxicos
- 2.2. Diagramas de Transición
- 2.3. Construcción de Thompson - Convertir una expresión regular a un AFN
- 2.4. Construcción de Subconjuntos - Convertir un AFN a un AFD
- 2.5. Construcción de un AFD a partir de una expresión regular (método del árbol)

Saber procedimental (habilidades y destrezas)

Reconoce los componentes del análisis de léxico.

Aplica los conceptos del análisis léxico en un proyecto de aplicación.

Saber actitudinal (conductas observables)

Escucha con atención las explicaciones del curso.

Participa activamente en las actividades de clase.

Trabaja de forma colaborativa con sus compañeros.

Realiza los trabajos asignados en tiempo.

Indicador de logro 2 (resultado):

Es capaz de utilizar analizadores léxicos en expresiones regulares dentro de un compilador, así como de forma independiente mediante regex.



COMPETENCIA 3

Reconoce las diferentes gramáticas que existen en lenguajes regulares y sus posibles implementaciones algorítmicas en analizadores sintácticos

Saber conceptual (contenido temático)

- 3.1. Comprobación del lenguaje generado
- 3.2. Supresión de ambigüedad
- 3.3. Eliminación de recursividad
- 3.4. Factorización a la izquierda
- 3.5. Análisis Sintáctico descendente (Top-Down)
- 3.6. Análisis descendente recursivo
- 3.7. Analizadores predictivos
- 3.8. Diagramas de transición
- 3.9. Análisis predictivo no recursivo
- 3.10. Construcción de tablas de análisis
- 3.11. Gramáticas LL(1)
- 3.12. Análisis sintáctico ascendente
- 3.13. Gramáticas LALR(1) o Construcción de tablas de parsers LALR

Saber procedimental (habilidades y destrezas)

- Aplica técnicas para eliminar defectos en las gramáticas mediante resolución de ejercicios.
- Reconoce los tipos de análisis sintáctico descendentes con un cuadro comparativo.
- Reconoce los tipos de análisis sintáctico ascendentes con un cuadro comparativo.
- Aplica el método de análisis sintáctico descendente con gramáticas LL(1) con resolución de ejercicios.
- Aplica el método de análisis sintáctico ascendente con gramática SLR, LR(1) y LALR(1) con resolución de ejercicios.
- Aplica unos de los métodos de análisis sintáctico en un proyecto de aplicación.

Saber actitudinal (conductas observables)

- Reconoce la importancia del análisis sintáctico en el proceso de compilación
- Escucha con atención las explicaciones del curso.
- Participa activamente en las actividades de clase.
- Trabaja de forma colaborativa con sus compañeros.
- Realiza los trabajos asignados en tiempo.

Indicador de logro 3 (resultado):

Aplica el proceso e importancia del análisis sintáctico en el proceso de compilación a través del desarrollo de un



proyecto de diseño de compilador simple.

COMPETENCIA 4

Implementa verificaciones de análisis semántico y type checkers en compiladores robustos y con tolerancia a errores

Saber conceptual (contenido temático)

- 4.1. Generación Gramáticas de atributo o Atributos sintetizados y heredados
- 4.2. Funciones semánticas
- 4.3. Comprobación de tipos
- 4.4. Comprobación estática y dinámica de tipos
- 4.5. Equivalencia de expresiones de tipos
- 4.6. Conversiones de tipos
- 4.7. Sobrecarga de funciones y operadores
- 4.8. Tabla de símbolos o Introducción
- 4.9. Atributos de símbolos

Saber procedimental (habilidades y destrezas)

Diferencia los tipos de atributos a nivel semántico y las gramáticas atribuidas.

Identifica lo aprendido sobre análisis semántico en una serie de ejercicios.

Resume la estructura y función de la tabla de símbolos.

Detalla la importancia del análisis semántico, comprobación de tipos y tabla de símbolos.

Aplica los conceptos de generación de atributos, comprobación de tipos y gestión de la tabla de símbolos en un proyecto de aplicación en equipos

Saber actitudinal (conductas observables)

Reconoce la importancia del análisis semántico en el proceso de compilación

Escucha con atención las explicaciones del curso.

Participa activamente en las actividades de clase.

Trabaja de forma colaborativa con sus compañeros.

Realiza los trabajos asignados en tiempo.

Indicador de logro 4 (resultado): Reconoce el proceso e importancia del análisis semántico en el proceso de compilación a través del desarrollo de un proyecto de diseño de compilador simple.



COMPETENCIA 5

Implementa un programa compilador hacia instrucciones de máquina virtual/ASM/tres direcciones como resultado de un árbol AST

Saber conceptual (contenido temático)

- 5.1. Traducción a una representación intermedia
- 5.2. Notación postfija
- 5.3. Árbol abstracto de análisis sintáctico
- 5.4. Código en tres direcciones
- 5.5. Preparación para la generación de código
- 5.6. Generación de directivas para la tabla de símbolos

Saber procedimental (habilidades y destrezas)

Conoce el método de generación de código en tres direcciones con resolución de ejercicios.

Resume los conceptos de la generación de código en un mapa conceptual.

Aplica los conceptos de generación de código en un proyecto de aplicación

Saber actitudinal (conductas observables)

Reconoce la importancia de la generación de código en el proceso de compilación

Escucha con atención las explicaciones del curso.

Participa activamente en las actividades de clase.

Trabaja de forma colaborativa con sus compañeros.

Realiza los trabajos asignados en tiempo.

Indicador de logro 5 (resultado): Distingue el proceso e importancia de la generación de código en el proceso de compilación a través del desarrollo de un proyecto de diseño de compilador simple realizado en equipos

Técnicas formativas	Procedimiento
Retroalimentación	Se proporciona la resolución de todos los problemas de las hojas de trabajo, laboratorios y exámenes realizados.
Diálogo socrático	Preguntas y respuestas orales a ejemplos y problemas que se realizarán lo largo de la secuencia de aprendizaje.
Exámenes cortos	Problemas de aplicación del tema seleccionado.
Trabajos en pequeños grupos para resolver dudas	Hojas de trabajo y laboratorios que se resuelven de forma colaborativa entre estudiantes.
Proyecto	Caso real donde se genera una solución en grupos utilizando compiladores como herramienta.



CALENDARIO DE REFERENCIA POR TEMAS

Fechas	Tema	Actividad de evaluación
08/07 – 12/07	Introducción y análisis léxico	Ejercicios prácticos.
15/07 – 19/07	Análisis sintáctico: Correcciones de gramática Análisis sintáctico descendente	Ejercicios prácticos.
22/07 – 26/07	Análisis sintáctico descendente predictivo	Ejercicios prácticos y laboratorio.
29/07 – 02/08	Análisis sintáctico	Ejercicios prácticos y laboratorio. Primer Examen Parcial.
05/08 – 09/08	Análisis sintáctico ascendente LR(0)	Ejercicios prácticos, entrega de enunciado de proyecto de aplicación.
12/08 – 23/08	Análisis sintáctico ascendente LR(1)	Ejercicios prácticos y proyecto de aplicación
26/08 – 06/09	Análisis sintáctico ascendente LALR(1)	Ejercicios prácticos y laboratorio
09/09 – 13/09	Análisis sintáctico y análisis semántico	Ejercicios prácticos y laboratorio
16/09 – 27/09	Análisis semántico y tabla de símbolos	Ejercicios prácticos y casos de aplicación Segunda evaluación parcial
30/09 – 04/10	Análisis semántico y comprobación de tipos	Ejercicios prácticos y laboratorio
07/10 – 11/10	Análisis semántico	Ejercicios prácticos y laboratorio
14/10 – 18/10	Introducción a la generación de código	Ejercicios prácticos.
21/10 – 25/11	Generación de código para VM	Ejercicios prácticos.
28/10 – 04/11	Generación de código para RISC	Presentación de Proyecto.
11/11 – 15/11		EXAMEN FINAL

REFERENCIAS BIBLIOGRÁFICAS

- Alfred V. Aho, Monica S. Lam, Ravi Sethi Jeffrey D. Ullman, Compiladores. Principios, Técnicas y Herramientas