



LENGUAJES FORMALES Y DE PROGRAMACIÓN
Primer Semestre 2020

I. INFORMACIÓN GENERAL

Código: 796	Créditos: 3
Escuela: Ciencias y Sistemas	Área: Ciencias de la Computación
El curso tiene laboratorio: Si	Categoría: Obligatorio
Horas magistrales a la semana: 2	Horas de laboratorio a la semana: 2
Prerrequisitos: 770 - Introducción a la Programación y Computación 1 795 - Lógica de Sistemas 960 - Matemática de Computo 1	
Postrequisitos: 772 - Estructuras de datos 777 - Organización de Lenguajes y Compiladores 1	

II. DISTRIBUCIÓN DE SECCIONES

Sección	Edificio	Salón	De:	A:	Lu	Ma	Mi	Ju	Vi	Sa	Catedrático
A-	T-3	113	7:10	8:50		X					VIVIAN DAMARIS CAMPOS GONZALEZ
A+	T-3	216	7:10	8:50		X					OTTO AMILCAR RODRIGUEZ ACOSTA
B-	T-3	214	7:10	8:50		X					ZULMA KARINA AGUIRRE ORDONEZ
B+	T-3	314	7:10	8:50		X					DAVID ESTUARDO MORALES AJCOT

III. DESCRIPCIÓN DEL CURSO

Este curso introduce al estudiante a los fundamentos conceptuales y teóricos matemáticos de los lenguajes de programación, como base para llevar los cursos de compiladores. Además, busca definir los modelos matemáticos asociados a la representación de los diferentes lenguajes, reconociendo los tipos de gramática para diseñar, resolver e implementar; específicamente las gramáticas regulares y las independientes del contexto mediante la práctica.

VII. CONTENIDO DECLARATIVO

UNIDAD 1: INTRODUCCIÓN A LOS LENGUAJES DE PROGRAMACIÓN Y FORMALES (8 PERIODOS)

Competencia específica: define, ubica el nivel, determina el paradigma de un lenguaje de programación, además determina los elementos de definición de los lenguajes de programación y formales y finalmente puede clasificar los tipos de gramática.

1. Lenguajes de programación (2 periodos)
 - 1.1. Definición
 - 1.2. Breve historia y generaciones
 - 1.3. Abstracciones de datos y control
 - 1.4. Niveles de lenguajes
 - 1.4.1. Alto nivel
 - 1.4.2. Bajo nivel
 - 1.4.3. Código de máquina
 - 1.5. Elementos de definición de lenguajes
 - 1.5.1. Sintaxis
 - 1.5.2. Semántica
 - 1.5.3. Pragmática

2. Paradigmas de los lenguajes de programación (2 periodos)
 - 2.1. Imperativo
 - 2.1.1. Definición y características
 - 2.1.2. Abstracciones de datos y control
 - 2.1.3. Procedimientos y funciones
 - 2.1.4. Ejemplos
 - 2.2. Orientado a objetos
 - 2.2.1. Definición y características
 - 2.2.2. Abstracción, encapsulamiento, herencia y polimorfismo
 - 2.2.3. Ejemplos
 - 2.3. Funcional
 - 2.3.1. Definición y características
 - 2.3.2. Recursividad
 - 2.3.3. Ejemplos
 - 2.4. Lógico
 - 2.4.1. Definición y características
 - 2.4.2. Predicados de primer orden
 - 2.4.3. Ejemplos

3. Lenguajes formales (2 periodos)
 - 3.1. Definición
 - 3.2. Alfabeto, símbolos, cadenas y reglas
 - 3.3. Tokens, patrones y lexemas
 - 3.4. Gramáticas en general
 - 3.5. Relación con el análisis léxico y sintáctico

4. Jerarquía de Chomsky (2 periodos)
 - 4.1. Clasificación de gramáticas según restricciones (Tipo 0, 1, 2, 3)
 - 4.2. Lenguajes recursivamente enumerables
 - 4.2.1. Máquinas de Turing
 - 4.3. Lenguajes dependientes del contexto
 - 4.3.1. Autómatas linealmente acotados
 - 4.4. Lenguajes independientes del contexto
 - 4.4.1. Autómatas a pila
 - 4.5. Lenguajes regulares
 - 4.5.1. Autómatas finitos
 - 4.5.2. Clasificación de los autómatas finitos

UNIDAD 2: LENGUAJES REGULARES (8 PERIODOS)

Competencia específica: diseña autómatas finitos, describe expresiones regular y utiliza gramáticas para el reconocimiento de cadenas.

1. Autómatas finitos deterministas (2 periodos)
 - 1.1. Definición
 - 1.2. Procesamiento
 - 1.3. Notación
 - 1.3.1. De conjuntos
 - 1.3.2. Diagramática
 - 1.3.3. Tabla de transiciones
 - 1.3.4. Ejemplos

2. Autómatas finitos no deterministas (2 periodos)
 - 2.1. Definición
 - 2.2. Clasificación según la presencia de épsilon
 - 2.3. Notación
 - 2.3.1. De conjuntos
 - 2.3.2. Diagramática
 - 2.3.3. Tabla de transiciones
 - 2.3.4. Ejemplos

3. Expresiones regulares (2 periodos)
 - 3.1. Definición
 - 3.2. Operadores
 - 3.3. Construcción
 - 3.4. Precedencia
 - 3.5. Ejemplos

4. Gramáticas regulares (2 periodos)
 - 4.1. Definición
 - 4.2. Reglas
 - 4.3. Tipos
 - 4.3.1. Por la izquierda
 - 4.3.2. Por la derecha
 - 4.3.3. Gramáticas lineales (varios símbolos)
 - 4.3.4. Ejemplos

UNIDAD 3: LENGUAJES INDEPENDIENTES DEL CONTEXTO (6 PERIODOS)

Competencia específica: diseña autómatas finitos, describe expresiones regular y utiliza gramáticas para el reconocimiento de cadenas.

1. Gramáticas independientes del contexto (2 periodos)
 - 1.1. Definición
 - 1.2. Derivaciones por la derecha e izquierda
 - 1.3. Árboles de derivación
 - 1.4. Ejemplos

2. Autómatas a pila (2 periodos)
 - 2.1. Definición
 - 2.2. Notación gráfica
 - 2.3. Descripción instantánea
 - 2.4. Aceptación por estado final y por pila vacía
 - 2.5. Ejemplos

3. Manejo de gramáticas independientes del contexto (2 periodos)
 - 3.1. Gramáticas ambiguas
 - 3.2. Eliminación de la ambigüedad
 - 3.3. Eliminación de recursividad por la izquierda
 - 3.4. Factorización
 - 3.5. Ejemplos

VIII. CONTENIDO PROCEDIMENTAL

UNIDAD 1:

- Descripción de conceptos
- Análisis diferentes paradigmas
- Clasificación lenguajes de programación
- Identificación tipos de gramática

UNIDAD 2:

- Elaboración autómatas finitos
- Descripción de expresiones regulares
- Construcción de gramáticas regulares

UNIDAD 3:

- Construcción de gramáticas independientes del contexto
- Elaboración autómatas a pila
- Manipulación las gramáticas para que sean eficientes

IX. CONTENIDO ACTITUDINAL

Para todas las unidades: se interesa por el contenido, participa activamente y promueve el aporte del grupo.

X. ACTIVIDADES DE LABORATORIO

El programa de laboratorio y los enunciados de las prácticas y proyectos los redactarán en conjunto los 4 auxiliares con las siguientes bases:

Práctica 1

Duración: 15 días

Fecha aproximada de entrega: 11 de febrero

Tema principal: recursividad para preparación de los cursos de compiladores

Lenguaje de programación: Python (consola)

Proyecto 1

Duración: 30 días

Fecha aproximada de entrega: 17 de marzo

Tema principal: implementación manual de los autómatas finitos deterministas y no deterministas mediante tabla de transiciones para reconocimiento de cadenas

Lenguaje de programación: C (consola)

Proyecto 2

Duración: 30 días

Fecha aproximada de entrega: 28 de abril

Tema principal: implementación manual de gramáticas regulares mediante el escáner de la misma para reconocimiento de cadenas

Lenguaje de programación: Java (gráfico)

Los lenguajes de programación seleccionados son los 3 más utilizados a nivel mundial, se puede utilizar cualquier índice, por ejemplo, el de Tiobe.

XI. INDICADORES DE LOGRO

- Describe conceptos de lenguajes de programación y formales
- Analiza adecuadamente los diferentes paradigmas
- Clasifica correctamente lenguajes de programación
- Identifica los tipos de gramática
- Elabora autómatas finitos de manera correcta
- Describe expresiones regulares
- Construye gramáticas regulares
- Construye gramáticas independientes del contexto
- Elabora autómatas a pila
- Manipula las gramáticas para que sean eficientes

XII. PRODUCTOS DE APRENDIZAJE

Según el Reglamento General de Evaluación y Promoción del Estudiante de la Universidad de San Carlos de Guatemala, la zona tiene valor de 75 puntos, la nota mínima de promoción es de 61 puntos y la zona mínima para optar a examen final es de 36 puntos.

3 parciales	45 puntos
Tareas, otras actividades	10 puntos
Laboratorio	20 puntos

Zona	75 puntos
Examen final	25 puntos

Nota final	100 puntos

XIII. CRITERIOS DE EVALUACIÓN

Conceptual declarativo: demuestra conceptos teóricos y matemáticos de los lenguajes formales mediante preguntas directas en clase.

Procedimental: aplica los conocimientos adquiridos de los lenguajes formales mediante hojas de trabajo, ejercicios, trabajo escritos y cursos en línea.

Actitudinal: participa activamente en la clase evaluado mediante la observación.

XIV. RECURSOS DIDÁCTICOS

- Pizarrón
- Marcadores y almohadilla
- Cañonera y presentaciones multimedia
- Libros de texto
- Documentos en Internet

XV. BIBLIOGRAFÍA Y RECURSOS

Libros de texto:

- Hopcroft, J., Motwani, R. y Ullman, J. (2005). Introducción a la teoría de autómatas, lenguajes y computación. España: Pearson Educación.
- Aho, M., Lam, R. y Ulmman, J. (2008). Compiladores: Principios, técnicas y herramientas. México: Pearson Educación.
- Louden, K., Lambert, K. (2011). Lenguajes de programación: principios y técnicas. USA: Cengage Learning.

Cursos tutoriales de Python, C y Java:

- <https://www.sololearn.com/Course/Python/>
- <https://www.sololearn.com/Course/C/>
- <https://www.sololearn.com/Course/Java/>