

Programación matemática 2

1. Descripción del Curso

Nombre: Programación Matemática 2 **Código:** M503
Prerrequisitos: M404 **Créditos:** 5
Profesor: Hugo García **Semestre:** Primero, 2019

Curso enfocado al teórico-práctico dedica al estudio de diseño de lenguajes formales de programación, que va desde la teoría de autómatas hasta las fases e implementación de un compilador para un lenguaje pequeño.

2. Competencias

2.1. Competencias generales

- 2.1.1 Capacidad para formular problemas, tomar decisiones e interpretar las soluciones en los contextos originales de los problemas.
- 2.1.2 Capacidad para utilizar las herramientas computacionales para plantear y resolver problemas.
- 2.1.3 Capacidad para comprender problemas, abstraer lo esencial de ellos y resolverlos.
- 2.1.4 Capacidad para extraer información cualitativa de datos cuantitativos.
- 2.1.5 Capacidad para detectar inconsistencias.

2.2. Competencias específicas

- a. El estudiante analiza requerimientos de lenguajes, diseña y construye autómatas para resolver dicho lenguaje.
- b. El estudiante implementa pequeñas porciones de código en lenguaje ensamblador.
- c. El estudiante reconoce los distintos tipos de lenguajes.
- d. El estudiante es capaz de hacer árboles léxicos.

3. Unidades

3.1. Vistazo a compiladores

Descripción: Introducción a los compiladores. Estructura de un compilador. Traducción.

Duración: 6 períodos de 50 minutos

Metodología: Los períodos de clase son magistrales.

Evaluación: Se evaluará a través de tareas semanales y preguntas directas en exámenes.

3.2. Análisis léxico

Descripción: Reconocimiento de palabras. Expresiones regulares. Autómatas finitos. Construcción de Thompson. Implementación de analizador léxico.

Duración: 22 períodos de 50 minutos

Metodología: Los períodos de clase son magistrales dando ejemplos de implementación de algoritmos en lenguaje de programación python.

Evaluación: Se evaluará a través de tareas semanales, preguntas directas y tres problemas en el primer examen parcial.

3.3. Análisis sintáctico

Descripción: Sintáxis. Analizador sintáctico Top-down. Analizador sintáctico Bottom-Up. Problemas prácticos. Optimización de una gramática. Reducción de tablas LR(1)

Duración: 30 períodos de 50 minutos

Metodología: Los períodos de clase son magistrales dando ejemplos de implementación de algoritmos en lenguaje de programación python.

Evaluación: Se evaluará a través de tareas semanales, preguntas directas y tres problemas en el segundo examen parcial.

4. Evaluación del curso

Los porcentajes asignados a cada uno de los elementos de la evaluación están de acuerdo con el Reglamento General de Evaluación y Promoción del Estudiante de la Universidad de San Carlos de Guatemala

Proyecto	24 puntos
Tareas semanales	10 puntos
Exámenes parciales	20 puntos
Práctica 1	5 puntos
Práctica 2	7 puntos
Práctica 3	9 puntos
Examen final	25 puntos
Total	100 puntos

5. Bibliografía

1. Cooper Keith. "Engineering a compiler". Segunda edición. USA 2012.
2. Fisher Charles. "Crafting a compiler". USA 2009.

<http://ecfm.usac.edu.gt/programas>