REPORTE DE PRÁCTICA NO. 2.1

NOMBRE DE LA PRÁCTICA: PRACTICA 0

ALUMNO: Adrian Omar Gonzalez Hernandez
Dr. Eduardo Cornejo-Velázquez

1. Introducción

El estudio de los lenguajes formales constituye un pilar fundamental en la teoría de la computación, ya que proporciona los modelos y herramientas necesarios para comprender cómo los sistemas informáticos procesan, interpretan y generan información. A través de los autómatas, las gramáticas y las expresiones regulares, es posible describir con precisión el comportamiento de programas, diseñar compiladores y validar datos en diferentes contextos.

2. Marco teórico

La teoría de Autómatas y Lenguajes Formales constituye un pilar fundamental en la ciencia de la computación, pues establece los principios matemáticos que permiten modelar, analizar y diseñar sistemas de procesamiento de información. Este marco conceptual ofrece las bases para comprender cómo los lenguajes formales describen conjuntos de cadenas y cómo los autómatas actúan como modelos abstractos de cómputo que pueden reconocer dichos lenguajes.

Un lenguaje formal se define como un conjunto de cadenas construidas a partir de un alfabeto finito. Las operaciones sobre lenguajes, como la unión, concatenación y clausura de Kleene, permiten generar nuevos lenguajes y describir patrones más complejos. En este contexto, las expresiones regulares representan una herramienta compacta para describir lenguajes regulares, los cuales son reconocidos por los autómatas finitos.

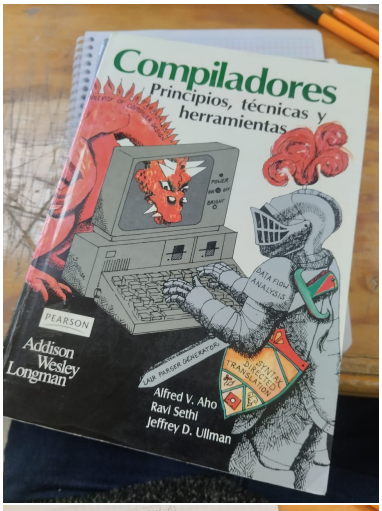
Los autómatas finitos deterministas (AFD) y no deterministas (AFND) son modelos matemáticos que procesan cadenas de entrada y determinan si pertenecen o no a un lenguaje. Aunque el AFND parece más flexible, ambos modelos poseen la misma potencia computacional, pudiendo transformarse entre sí. Variaciones como los autómatas con transiciones facilitan la construcción de autómatas a partir de expresiones regulares. Además, la minimización de autómatas permite simplificar los modelos sin alterar el lenguaje que reconocen. La teoría de lenguajes se enriquece con resultados como el Lema de Bombeo, herramienta que permite demostrar si un lenguaje no es regular. Sin embargo, no todos los lenguajes pueden representarse mediante autómatas finitos; por ello surgen modelos más complejos, como las gramáticas formales y los autómatas de pila (PDA). Estos últimos reconocen los lenguajes libres de contexto, que incluyen estructuras esenciales en la programación, como la correcta anidación de paréntesis.

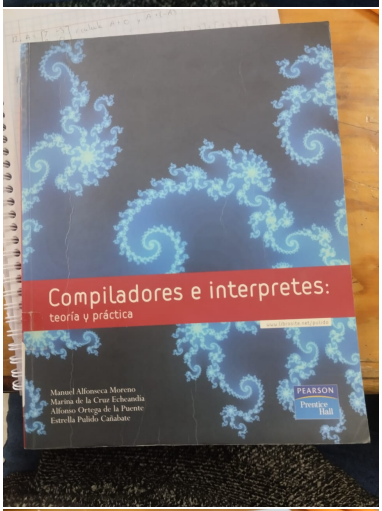
La jerarquía de Chomsky organiza los lenguajes en cuatro niveles de expresividad:
Tipo 3: Lenguajes Regulares.
Tipo 2: Lenguajes Libres de Contexto.
Tipo 1: Lenguajes Sensibles al Contexto.
Tipo 0: Lenguajes Recursivamente Enumerables.
Cada nivel tiene asociado un modelo computacional que lo reconoce, mostrando el progreso de lo simple a lo más general en el estudio del cómputo.

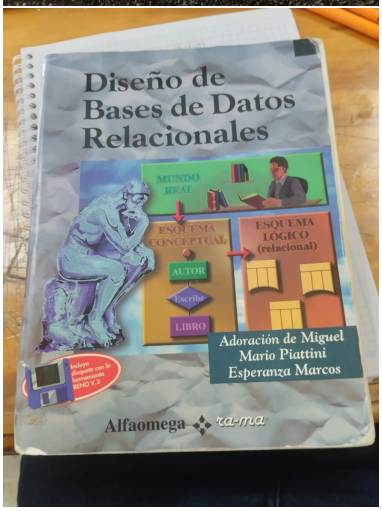
3. Herramientas empleadas

1. Editor LaTeX: Overleaf
2. Biblioteca digital

4. Desarrollo







5.Conclusiones

El estudio de los autómatas y lenguajes formales permitió comprender cómo los modelos matemáticos pueden describir y reconocer patrones en los lenguajes, sentando las bases para áreas esenciales de la informática como el diseño de compiladores, el análisis léxico y el procesamiento de datos. La práctica reforzó la importancia de los conceptos teóricos al observar sus aplicaciones en ejemplos concretos, así como la utilidad de las herramientas digitales.

References

[1] Codemath (s/f). Autómatas y Lenguajes Formales DESDE CERO. *Youtube*. Recuperado el 21 de septiembre de 2025, de `http://www.youtube.com/playlist?list=PLyRNgg3I27WiOZqDamrZon3QDPZMlZ5P4`.

[2] Aho, A. V.; Sethi, R.; Ullman, J. D. (**2007**). Compiladores: Principios, técnicas y herramientas (2a ed.). *Pearson Addison Wesley*.

[3] Alfonseca Moreno, M.; De la Cruz Echeandía, M.; Ortega de la Puente, A.; Pulido Cañabate, E. (**2004**). Compiladores e intérpretes: teoría y práctica. *Pearson Prentice Hall*.

[4] Piattini, M.; Marcos, E. (**2001**). Diseño de bases de datos relacionales. *Ra-Ma*.