

Plano da Disciplina: Teoria da Computação I

1. Desenho da Disciplina

- **Nome:** Teoria da Computação I
 - **Curso:** Licenciatura em Ciência da Computação, Engenharia de Software ou áreas correlatas
 - **Ano/Semestre:** 1º Ano / 2º Semestre
 - **Modalidade:** Presencial
 - **Carga Horária Semanal:** 5 horas
 - **Carga Horária Total:** 80 horas
-

2. Regime

- **Regime:** Semestral
 - **Frequência mínima:** 75% de presença nas aulas teóricas e práticas
 - **Avaliação contínua** com provas, exercícios e projetos computacionais
-

3. Posição no Curso

- **1º Ano, 2º Semestre**
 - Disciplina fundamental para **bases teóricas da computação e linguagens formais**
-

4. Horário Semanal

- **2 horas teóricas** (fundamentos matemáticos e modelos computacionais)
 - **1 hora teórico-prática** (exemplos aplicados e resolução de problemas)
 - **2 horas práticas (laboratórios)** (simulação e implementação computacional)
-

5. Carga Horária Total

- **Total:** 80 horas
 - **Horas Teóricas:** 32h
 - **Horas Teórico-Práticas:** 16h
 - **Horas Práticas (Laboratórios):** 32h
-

6. Precedências Obrigatórias

- **Não há disciplinas precedentes obrigatórias.**
-

7. Objetivos Gerais

- 1 Apresentar conceitos fundamentais de **teoria da computação, linguagens formais e lógica matemática**.
 - 2 Desenvolver habilidades para trabalhar com **modelos computacionais clássicos** como **autômatos finitos, máquinas de Turing e autômatos celulares**.
 - 3 Entender e aplicar conceitos de **linguagens formais, gramáticas e recursos computacionais**.
 - 4 Explorar os **limites computacionais de algoritmos e problemas**.
 - 5 Introduzir a **computação como um conceito físico-matemático**, além de sua abordagem puramente teórica.
-

8. Competências a Desenvolver

- ✓ Compreender os fundamentos da **Teoria da Computação e Linguagens Formais**.
 - ✓ Aplicar conceitos de **autômatos e gramáticas** na modelagem computacional.
 - ✓ Construir e analisar **Máquinas de Turing e sua relação com computabilidade**.
 - ✓ Resolver problemas usando **autômatos celulares e computação paralela**.
 - ✓ Simular e implementar modelos computacionais em **JFLAP, Python e outras ferramentas**.
-

9. Conteúdo Temático e Bibliografia

Unidade 1: Linguagens Regulares e Autômatos Finitos (10h)

Michael Sipser - Introduction to the Theory of Computation (2006)

- Gramáticas e Linguagens
- Linguagens Regulares e Expressões Regulares
- Autômatos de Estados Finitos e suas variantes

Laboratório: Implementação de autômatos finitos no JFLAP

Unidade 2: Autômatos Finitos com Saída (10h)

Lewis & Papadimitriou - Elementos de Teoria da Computação (2006)

- Máquinas de Mealy e Moore
- Aplicações de autômatos finitos
- Minimização de autômatos

Laboratório: Criação de autômatos finitos para reconhecimento de padrões

Unidade 3: Linguagens Livres de Contexto e Autômatos de Pilha (15h)

Hopcroft, Ullman & Motwani – Introdução à Teoria de Autômatos, Linguagens e Computação (2003)

- Gramáticas Livres de Contexto
- Lema do Bombeamento para Linguagens Livres de Contexto
- Autômatos de Pilha e reconhecimento de linguagens

Laboratório: Implementação de autômatos de pilha para análise sintática

Unidade 4: Linguagens Sensíveis ao Contexto e Autômatos Limitados Linearmente (10h)

Nicoletti & Hruschka – Fundamentos da Teoria dos Grafos para Computação (2010)

- Gramáticas Sensíveis ao Contexto
- Autômatos Limitados Linearmente

Laboratório: Construção de gramáticas para linguagens sensíveis ao contexto

Unidade 5: Máquinas de Turing e Computabilidade (15h)

Michael Sipser – Introduction to the Theory of Computation (2006)

- Modelos de Máquina de Turing
- Decidibilidade e Indecidibilidade
- Conjunto de Aceitação de uma Máquina de Turing

Laboratório: Simulação de Máquinas de Turing no JFLAP

Unidade 6: Autômatos Celulares e Computação Natural (10h)

Max Garzon – Models of Massive Parallelism: Analysis of Cellular Automata and Neural Networks (1995)

- Conceito de autômatos celulares
- Regras de evolução e comportamento emergente
- Aplicações em computação natural

Laboratório: Simulação de autômatos celulares no Python

Unidade 7: Projeto Final Integrado (10h)

Baseado nas competências adquiridas ao longo do semestre

- Desenvolvimento de um **projeto de modelagem computacional**
- Aplicação dos conceitos de autômatos e computabilidade

10. Metodologia de Trabalho

Aulas teóricas para introdução aos conceitos matemáticos e computacionais

Exercícios práticos para aplicação de conceitos formais

Uso de software (JFLAP, Python) para simulação computacional

Projetos individuais e em grupo para aprofundamento dos conceitos

11. Critério de Avaliação

- ✓ 40% - Trabalhos práticos e atividades laboratoriais
 - ✓ 30% - Projeto Final Integrado
 - ✓ 20% - Participação e atividades complementares
 - ✓ 10% - Prova escrita e técnica
-

12. Bibliografia Básica

Michael Sipser – *Introduction to the Theory of Computation*, 2ª Ed., Thomson Course Technology, 2006

Hopcroft, J. E., Ullman, J. D., Motwani, R. – *Introdução à Teoria de Autômatos, Linguagens e Computação*, Editora Campus, 2003

Lewis, H. R., Papadimitriou, C. H. – *Elementos de Teoria da Computação*, 2ª Edição, Bookman, 2006

13. Bibliografia Complementar

Max Garzon – *Models of Massive Parallelism: Analysis of Cellular Automata and Neural Networks*, Springer, 1995

Nicoletti, M. C., Hruschka, E. R. – *Fundamentos da Teoria dos Grafos para Computação*, EdUfscar, 2010

Paulo Blauth Menezes, Edward Hermann Haeusler – *Teoria das Categorias para Ciência da Computação*, Editora Sagra Luzzatto