



## PLANO ANALÍTICO DA UNIDADE CURRICULAR

Faculdade/Escola/Centro Engenharia

Departamento de Engenharia Electrotécnica Ano lectivo de 2024 Semestre: I

Curso (s) : Licenciatura em Engenharia Informática

Unidade Curricular: Compiladores

Nome do (s) docente (s) (Regente): Ruben Manhiça

Nome do (s) Docente (s) (Assistente): Cristialino Maculuve

Nome do técnico: \_\_\_\_\_

Nome do monitor: \_\_\_\_\_

Regime: Diurno

Horas e créditos:

	Práticas	Teóricas	Teórico-Prática	Seminários	Avaliação	Outras	Total
Horas de contacto directo por semestre	18	32	14	0	10		74
Horas de contacto directo por semana	2	2		0			4
Horas de estudo independente por semestre	0	32	48	0			80
Créditos <sup>1</sup>							

### I. INTRODUÇÃO

#### Características gerais da UC

Segundo o plano de estudos em vigor para o curso de licenciatura em Engenharia Informática a disciplina de compiladores ocorre no primeiro semestre do 4o ano, tanto para o período laboral como para o pós-laboral. Por definição segundo APPEL e PALSBERG (2004) um compilador é um programa de computador (ou um grupo de programas) que, a partir de um código fonte escrito em uma linguagem (Geralmente em uma linguagem de alto nível) cria um programa semanticamente equivalente, porém escrito em outra linguagem. Nesta disciplina se espera que os estudantes possam

---

<sup>1</sup> Na UEM, o crédito académico corresponde a um total de 30 horas de trabalho.

conhecer a importância dos compiladores como um dos pontos-chave no desenvolvimento de sistemas e tecnologias de informação, compreender as fases de compilação de linguagens, em especial das linguagens imperativas e Orientadas a Objetos e acima de tudo compreender e aplicar as estruturas de dados e os algoritmos principais usados na implementação de compiladores.

#### Objectivos:

Esta disciplina tem como objectivo capacitar os estudantes com conhecimentos que os permitam implementar as principais fases da execução de um compilador em uma linguagem de alto nível, usando os diversos recursos e bibliotecas disponíveis.

#### Resultados de aprendizagem

Espera-se que ao fim desta disciplina o estudante possa:

- Compreender as fases de compilação de linguagens, em especial das linguagens imperativas e OO;
- Especificar a sintaxe e semântica de uma linguagem de programação;
- Compreender e utilizar as estruturas de dados e os algoritmos principais usados na implementação de compiladores.

## **II. ESTRATÉGIAS DE DE ENSINO E DE APRENDIZAGEM**

### a) Tipo de aulas e formas de leção

Nas aulas teóricas serão apresentados os conceitos gerais e exemplos. Nas aulas práticas e laboratoriais serão realizados os exercícios e demonstrações para a consolidação das matérias teóricas. Durante o estudo individual o estudante desenvolverá as habilidades por meio de leitura e resolução de problemas práticos, fazendo as pesquisas individualmente e/ou em grupo.

### b) Actividades de frequência obrigatória

- a. Aulas praticas
- b. Testes
- c. Laboratórios
- d. Seminários

## **III. ESTRATÉGIAS DE AVALIAÇÃO**

O processo de avaliação desta disciplina irá obedecer as diretrizes estipuladas no capítulo 7 do regulamento pedagógico em vigor na Universidade Eduardo Mondlane. De forma a permitir maior envolvimento dos estudantes e como forma de permitir maior oportunidade aos mesmos a avaliação será realizada através de 3 testes escritos, diversos TPC's (feitos individualmente ou em grupo), 1 trabalho de pesquisa envolvendo desenvolvimento de um protótipo sobre um tema apresentado e também a assiduidade do estudante as aulas.

A nota de frequência será calculada na base da fórmula:

$$NF = 0.6*MT + 0.2*TP + 0.2*E$$

Onde:

- **NF:** Nota de Frequência;
- **MT:** Media dos Testes;
- **TP:** Trabalho Prático (de investigação);
- **E:** Envolvimento nas aulas, pela fórmula:

$$E = 0.6*TPC + 0.25*Presenças$$

**NB:** Sobre a participação do estudante as aulas, prevalecem todos os pressupostos do artigo 37 do regulamento pedagógico da Universidade Eduardo Mondlane.

#### IV. TEMÁTICAS

Nº	Temas	Horas				
		Teóricas	Práticas	S	EI	Total
01	1. Introdução. Fases da compilação e estrutura típica de um compilador.	2	2		5	9
02	2. Análise léxica. Expressões regulares e autómatos finitos.	2	2		5	9
03	3. Análise sintática. Fundamentos teóricos. Gramáticas. Algoritmos para análise sintática. Tratamento de erros.	4	6		13	23
04	4. Análise semântica. Verificação de tipos.	4	4		10	18
05	5. Ambientes de execução. Organização de memória e mecanismos para a passagem de parâmetros.	8	6		17	31
06	6. Técnicas de geração de código intermédio	6	6		15	27
07	7. Técnicas de geração de código final e de optimização	6	6		15	27

## V. CALENDÁRIO DAS AULAS E DAS AVALIAÇÕES

No. da aula ou avaliação	Data	Tema da aula	Tipo	Material de apoio para aula
1		Apresentação da disciplina; Considerações introdutórias sobre a metodologia de ensino e avaliação; Introdução a compiladores (História, Tradutores, Interpretadores, Montadores e Pré Processadores)	T	Slide 0 – Compiladores - Apresentação do plano temático; Slide 1 - Introdução aos Compiladores;
2		Aula Prática: Resolução de Exercícios	P	FICHA DE EXERCICIOS
3		Compilação vs Interpretação; Estrutura típica de um compilador; Etapas de compilação;	T	Slide 2 – Etapas de Compilação;
4		Aula Prática: Resolução de Exercícios	P	FICHA DE EXERCICIOS
5		Introdução ao Analisador Léxico	T	Slide 3-Introdução ao Analisador Léxico
6		Aula Prática : Resolução de Exercícios	P	FICHA DE EXERCICIOS
7		AFD e AFND	T	Slide 4 AFD e AFND
8		Aula Prática : Resolução de Exercícios	P	FICHA DE EXERCICIOS
9		Expressões Regulares Considerações sobre o Teste 1	T	Slide 5-Expressões Regulares
10		<b>TESTE 1</b>		
11		Exercícios sobre Expressões Regulares	P	FICHA DE EXERCICIOS
12		<b>FERIADO 7 DE ABRIL / SEXTA SANTA</b>	P	FICHA DE EXERCICIOS
13		Exercícios sobre Expressões Regulares	T	Slide 6-Introdução ao Analisador Sintático
14		Introdução ao Analisador Sintático		
15		Tipos de Analisadores Hierarquia de Chomsky	T	Slide 7-Tipos de Analisadores Hierarquia de Chomsky
16		Exercícios Analisador Sintático		
17		Tratamento de Erros Sintáticos Autômato de Pilha	T	Slide 8-Tratamento de Erros Sintáticos e Autômato de Pilha
18		Exercícios sobre Tratamento de Erros Sintáticos Autômato de Pilha	P	FICHA DE EXERCICIOS
19		Distribuição de Temas do Projecto Considerações sobre o Teste 2	S	
20		<b>TESTE 2</b>		
21		Introdução a Análise Semântica	T	Slide 9-Introdução a Análise Semântica
22		<b>FERIADO 1 DE MAIO</b>	P	FICHA DE EXERCICIOS
23		Exercícios Introdução a Análise Semântica Tabela de Símbolos	T	Slide 10-Tabela de Símbolos
24		Exercícios Tabela de Símbolos	P	FICHA DE EXERCICIOS
25		Síntese, Geração de código intermedio e final e otimização	T	Slide 11- Síntese, Geração de código intermedio e final e otimização
26		Exercícios sobre Síntese, Geração de código intermedio e final e otimização	P	FICHA DE EXERCICIOS
27		Acompanhamento dos Projectos da disciplina	P	
28		Acompanhamento dos Projectos da disciplina	P	
29		Acompanhamento dos Projectos da disciplina	P	
30		<b>TESTE 3</b>		
31		DEFESAS	S	
32		DEFESAS	S	

## **VI. BIBLIOGRAFIA E RECURSOS**

- [1] AHO, Alfred V., M. SETHI, Ravi, ULLMAN, Jeffrey J., Compilers: Principles, Techniques, and Tools, 2nd Edition, Addison Wesley 2007.
- [2] COOPER, Torczon. Engineering a Compiler , San Francisco, Morgan Kaufmann, 2003;
- [3] NETO, João José. Introdução à Compilação , Rio de Janeiro: LTC, 1987
- [4] WATT, David A., BROWN, Deryck F., Programming Language Processors in Java, Harlow, England, Prentice Hall, 2000