

CENTRO DE ENSINO	CURSO
CETEC	ENGENHARIA DE COMPUTAÇÃO

### COMPONENTE CURRICULAR

CÓDIGO	NOME	TURMA
CET525	COMPILEADORES	01

#### DOCENTE

TASSIO FERREIRA VALLE

#### PRÉ-REQUISITO(S)

CET224 ESTRUTURAS DE DADOS

CET518 LINGUAGENS FORMAIS E AUTÔMATOS

#### CO-REQUISITO(S)

N/A

#### NATUREZA

OBRIGATÓRIA

CARGA HORÁRIA					
TÉORICA	PRÁTICA	TOTAL	EAD	AÇÃO DE EXTENSÃO CURRICULARIZADA	
68	-	68	-	-	-

#### EMENTA

Linguagens-fonte, objeto, de alto-nível e de baixo-nível. Compilação e interpretação. Processadores de linguagens de programação. Máquinas reais e virtuais. Especificação de linguagens de programação. Bootstrapping. Análise sintática. Análise de contexto. Ambientes de execução. Geração de código. Otimização de código independente de máquina. Otimização de código dependente de máquina.

#### OBJETIVOS

- Estimular a capacidade de resolver os problemas relacionados à componente curricular
- Fomentar o trabalho em equipe, compromisso com prazos estipulados e qualidade nos resultados
- Apresentar os conceitos mais recentes da componente curricular ao aluno, destacando as melhores práticas utilizadas no mercado

#### CONTEÚDO PROGRAMÁTICO

## **1. Introdução aos Compiladores**

- Linguagens-fonte e linguagens-objeto
- Compilação × interpretação
- Arquitetura geral de um compilador

## **2. Processadores de Linguagens e Máquinas**

- Processadores de linguagens de programação
- Máquinas reais e máquinas virtuais
- Especificação de linguagens

## **3. Bootstrapping**

- Conceito e importância
- Estratégias de autocompilação

## **4. Análise Léxica**

- Tokens, padrões e lexemas
- Expressões regulares
- Autômatos finitos
- Construção do analisador léxico

## **5. Análise Sintática**

- Gramáticas livres de contexto
- Parsing LL e LR
- Árvores sintáticas e AST
- Tratamento de erros sintáticos

## **6. Análise Semântica**

- Tipagem e coerência
- Verificação de contexto
- Tabelas de símbolos

## **7. Ambientes de Execução**

- Pilhas, heaps e frames
- Gerenciamento de memória
- Escopo estático e dinâmico

## **8. Representação Intermediária**

- IR linear e baseada em árvore
- Three-address code
- Transformações estruturais básicas

## 9. Geração de Código

- Seleção de instruções
- Alocação de registradores
- Geração de código final

## 10. Otimizações de Código

- Otimizações independentes de máquina
- Otimizações dependentes de máquina
- Otimizações de fluxo de controle e dados

## 11. Compiladores Reais e Ferramentas

- Uso de ferramentas (Lex/Flex, Yacc/Bison, ANTLR)
- Arquitetura de compiladores reais (GCC, Clang)

## METODOLOGIA DE ENSINO

A metodologia adotada nesta disciplina será centrada no aluno, promovendo o protagonismo e incentivando a participação ativa no processo de aprendizagem. O conteúdo será abordado por meio de uma combinação de práticas ativas, como aprendizagem baseada em projetos, estudos de caso e discussões em grupo.

- 1. Aprendizagem Baseada em Projetos (PBL):** Os alunos serão desafiados a construir parte de um compilador simples ao longo do curso. Em cada etapa, eles deverão aplicar os conceitos aprendidos, como análise léxica, sintática e semântica, além da geração e otimização de código.
- 2. Estudos de Caso:** Cada aula terá um estudo de caso real onde os alunos investigarão como técnicas de compilação são aplicadas em compiladores modernos, como GCC e Clang. Isso permitirá que os alunos vejam a teoria em ação e compreendam melhor o contexto prático.
- 3. Discussões em Grupo:** Serão promovidas discussões semanais sobre tópicos controversos e desafiadores, como os desafios na otimização de código dependente de máquina ou as decisões de design dos compiladores.
- 4. Aprendizagem Colaborativa:** O trabalho em grupo será incentivado em várias etapas do projeto. Os alunos colaborarão no desenvolvimento de diferentes partes do compilador, discutindo e resolvendo problemas em conjunto.
- 5. Atividades Práticas e Laboratórios:** Além das aulas teóricas, os alunos terão sessões práticas em laboratório, onde poderão codificar e testar suas próprias implementações, com orientação contínua do professor.
- 6. Autoavaliação e Avaliação por Pares:** Para reforçar o aprendizado reflexivo, os alunos serão convidados a autoavaliar seu progresso no projeto e a avaliar o trabalho dos colegas, proporcionando uma compreensão mais profunda do processo de desenvolvimento de compiladores.

A ideia é criar um ambiente de aprendizado ativo, onde os alunos não apenas consomem conteúdo, mas aplicam o conhecimento em projetos reais, discutem questões técnicas de forma colaborativa e se envolvem ativamente na construção do conhecimento.

## AVALIAÇÃO DA APRENDIZAGEM

A avaliação constará de:

- Projeto prático;

$$\text{Média Final} = ((\text{Trabalho - Parte 1} * 50) + (\text{Trabalho - Parte 2} * 50)) / 100$$

## BIBLIOGRAFIA

**Bibliografia Básica do Componente Curricular**

1. Watt D.A; Brown, D. F. Programming Language Processors in Java – Compilers and Interpreters. Pearson Education, 2000.
2. Aho, A. V. et al, Compiladores – Princípios, Técnicas e Ferramentas. 2a ed. Pearson Education, 2007.
3. Louden, K.C. Compiladores – Princípios e Práticas. Cengage Learning, 2004.

**Bibliografia Complementar do Componente Curricular**

1. LEWIS, Harry R. & PAPADIMITRIOU, Christos H. Elementos de Teoria da Computação. 2.ed. Porto Alegre, Bookman,2000.
2. Price, A.M.A.; Toscani, S.S. Implementação de Linguagens de Programação – Compiladores. Bookman, 2008.
3. Ricarte I. Introdução à Computação. Elsevier. 2008.
4. Delamaro, M.E. Como construir um Compilador – Utilizando Ferramentas Java. Novatec, 2004
5. Grune, H.D. et al. Projeto Moderno de Compiladores – implementação e Aplicações. Elsevier, 2001.

**CRONOGRAMA DE ATIVIDADES**

DATAS	ATIVIDADES PROGRAMADAS
01/02/2026	Introdução a Compiladores e Linguagens de Programação
02/02/2026	Linguagens Fonte e Objeto
03/02/2026	Processadores de Linguagens
04/02/2026	Máquinas Reais e Virtuais
05/02/2026	Especificação de Linguagens de Programação
08/02/2026	Bootstrapping em Compiladores
09/02/2026	Análise Léxica
10/02/2026	Análise Sintática
11/02/2026	Análise Semântica e Análise de Contexto
12/02/2026	Ambientes de Execução
15/02/2026	Geração de Código
16/02/2026	Feriado
17/02/2026	Feriado
18/02/2026	Feriado
19/02/2026	Otimização de Código Independente de Máquina
22/02/2026	Otimização de Código Dependente de Máquina
23/02/2026	Ferramentas de Compilação
24/02/2026	Compiladores de Linguagens Reais
25/02/2026	Desafios e Futuro dos Compiladores
26/02/2026	Projeto Final

**USO DE ANIMAIS NAS ATIVIDADES DE ENSINO**

Não se aplica.

**AÇÃO DE EXTENSÃO CURRICULARIZADA**

Não se aplica.

<b>Assinatura do Docente Responsável</b>	28/11/2024
<hr/> <b>Docente</b>	

NI

<b>Data de Aprovação em Reunião do Colegiado do Curso</b>	
<hr/> <b>Coordenador(a)</b>	
<b>Data de Homologação em Reunião do Conselho Diretor do Centro</b>	
<hr/> <b>Presidente do Conselho Diretor do Centro de Ciências Exatas e Tecnológicas</b>	