



PLANO DE APRENDIZAGEM

1. DADOS DE IDENTIFICAÇÃO	
Curso: Bacharelado em Sistemas de Informação	
Disciplina: Linguagens Formais e Compiladores	Código: SIF5N191
Professor: Me. Igor Peterson Oliveira Santos	e-mail: igor.santos@fasete.edu.br
CH Total: 40h	Créditos: 02
Pré-requisito(s): -	
Período: V	Ano: 2019.2

2. EMENTA:

Processo de Compilação e Componentes de um Compilador; Analisadores Léxicos; Analisadores Sintáticos; Analisadores Semânticos e Autômatos.

3. COMPETÊNCIAS E HABILIDADES DA DISCIPLINA:

Identificar, selecionar, configurar e gerenciar tecnologias da informação nas organizações. O aluno estará apto a configurar e construir compiladores.

4. OBJETIVO GERAL DA APRENDIZAGEM:

Conhecer a teoria das linguagens formais visando sua aplicação na especificação de linguagens de programação e na construção de compiladores.

5. CONTEÚDOS:

5.1. 1ª ETAPA

CONTEÚDOS PRESENCIAIS (20h)

1. Compiladores e Interpretadores (02 h)
 - Definições básicas de compiladores e interpretadores.
 - Etapas de tradução de programas.
2. Análise Léxica (08 h)
 - Tradução simples dirigida por sintaxe.
 - Reconhecimento de tokens.
 - Gerador de analisador léxico.
 - Expressões regulares e autômatos.
 - Gramática livre de contexto.
3. Análise Sintática (08 h)
 - Análise sintática descendente e ascendente.
 - Análise LR.
 - Gerador de analisador sintático.



5.2. 2ª ETAPA

CONTEÚDOS PRESENCIAIS (20h)

1. Tabelas de símbolos (**02 h**)
 - Estrutura.
 - Atributos.
 - Gerenciamento da tabela de símbolos.
 - Técnicas de implementação da tabela de símbolos.
2. Análise Semântica (**08 h**)
 - Tipos de dados e verificação de tipos.
 - Analisador semântico.
3. Recuperação de erro (**06 h**)
 - Recuperação de erros na análise LL.
 - Recuperação de erros na análise de precedência de operadores.
 - Recuperação de erros na análise LR.
4. Geração de código (**04 h**)
 - Geração de código intermediário
 - Variantes das árvores de sintaxe.
 - Código de três endereços.
 - Tradução de expressões.
 - Verificação de tipo.
 - Fluxo de controle.

6. METODOLOGIA DO TRABALHO:

Os procedimentos a serem adotados serão os seguintes: aulas expositivas intercaladas com aulas práticas onde serão apresentados e construídos compiladores utilizando os conceitos apresentados; projeto de desenvolvimento de um compilador.

2ª ETAPA

1) Atividades Processuais (10,0 pontos)

Serão divididas em 2 atividades, como segue:

a. Práticas de Programação (5,0 pontos)

Serão propostos problemas a serem resolvidos individualmente ou em duplas sobre os conteúdos abordados em sala para a construção de um compilador. O aluno poderá realizar estudos específicos e apresentar as soluções para os problemas indicados pelo professor.

b. Sala invertida (5,0 pontos)

A proposta da sala invertida é prover aulas menos expositivas, mais produtivas e participativas, capazes de engajar os alunos no conteúdo e melhor utilizar o tempo e conhecimento do professor. Sendo assim, será proposto para os alunos, por meio de pesquisas e/ou leituras extraclasse, o estudante terá acesso prévio do conteúdo curricular de compiladores e estudar antes de ir para a sala de aula, ocasião em que discutirá com colegas e professor os assuntos já vistos em casa.



2) Avaliação Institucional (10,0 pontos)

A avaliação escrita será composta por questões objetivas e subjetivas, versando sobre todos os temas discutidos na 1ª Etapa da disciplina. A avaliação será concebida no mesmo modelo estrutural de elaboração de questões utilizados nas provas do ENADE, suas regras e orientações estarão expostas no campo de instruções. A avaliação será individual, e aplicada de acordo com o calendário acadêmico. Caso ocorram alterações no Calendário Acadêmico de 2019, esta data poderá ser alterada, e devidamente comunicada aos estudantes.

2ª ETAPA

3) Aprendizagem Entre Equipes ou Dupla (10,0 pontos)

Serão realizados uma sequência de trabalhos em duplas ou equipes para que os mesmos resolvam os desafios e trabalhem juntos, o que pode ser benéfico na busca pelo conhecimento.

4) Avaliação Institucional (10,0 pontos)

Esta avaliação segue sob as mesmas diretrizes do item 2) Avaliação Institucional.

7. RECURSOS:

<input type="checkbox"/> Sala (comum)	<input type="checkbox"/> Sala Configuração Flexível	<input checked="" type="checkbox"/> Laboratório(s)
<input type="checkbox"/> Práticas em Campo	<input checked="" type="checkbox"/> Kit multimídia	<input type="checkbox"/> Outros (informar)

8. SISTEMA DE AVALIAÇÃO:

1ª ETAPA

- a) Atividades Processuais (10,0 pontos)
- b) Avaliação Institucional (10,0 pontos)

2ª ETAPA

- a) Aprendizagem Entre Equipes ou Dupla (10,0 pontos)
- b) Avaliação Institucional (10,0 pontos)

Observações:

- **Média das Etapas:** será igual a média aritmética das avaliações nas respectivas etapas.
- **Detalhes das atividades no item 6. METODOLOGIA DO TRABALHO**

9. ATENDIMENTO EXTRA CLASSE:

Em caráter complementar, o professor oferece atendimento, diariamente, através do endereço eletrônico: igor.santos@faseite.edu.br



10. BIBLIOGRAFIA BÁSICA:

- DELAMARO, M. E. **Como Construir Um Compilador**. São Paulo: Editora Novatec, 2010. 307 P. Isbn 85-7522-055-1.
- AHO, A. V., SETHI, R., ULLMAN, J. D. **Compiladores: Princípios, Técnicas E Ferramentas**. 2ª Ed. São Paulo: Pearson Education, 2008. 634 P. Isbn 978-85-88639-24-9.
- COOPER, Keith D. **Construindo Compiladores**. 2ª Ed. Rio De Janeiro: Elsevier, 2014. 656 P. Isbn 978-85-352-5564-5.
- PRICE, A. M. A. **Implementação De Linguagens De Programação: Compiladores**. 3ª. Porto Alegre: Sagra-Luzzatto, 2005. 212 P. Isbn 85-241-0639-5.

11. BIBLIOGRAFIA COMPLEMENTAR:

- RAMOS, M. V. M., NETO, J. J., VEJA, I. S., **Linguagens Formais: Teoria, Modelagem e Implementação**. Ed. Bookman, 2009.
- HOPCROFT, J. F., ULLMAN, J. D., MOTWANI, R. **Introdução à Teoria de Autômatos, Linguagens e Computação**, Tradução da segunda edição Americana, Elsevier Editora Ltda, 2003.
- MENESES, P. B. **Linguagens Formais e Autômatos**, Ed. Sagra Luzzato, 5. edição, 2005.
- FISHER, C. N., CYTRON, R. K., LeBLANNC Jr., R. J.. **Crafting a Compiler**. Ed. Addison Wesley, 2009.
- HOPCROFT, J. E., ULLMAN, J. D. **Formal Languages and Their Relations to Automata**. Addison-Wesley, 1969.
- HOPCROFT, J. F., ULLMAN, J. D.. **Introduction to Automata Theory, Languages and Computation**. Ed. Addison-Wesley, 1979.
- DIVERIO, T. A., MENEZES, P. B., Teoria da Computação – Máquinas Universais e Computabilidade. Ed. Sagra Luzzatto, Porto Alegre, 1999
- WOOD, D. Theory of Computation. Ed. John Wiley & Sons, 1987.

12. LEITURA COMPLEMENTAR

13. INFORMAÇÕES COMPLEMENTARES:

O aluno que faltar a prova escrita da 1ª chamada terá o direito de fazer uma prova de Segunda Chamada, cujo conteúdo abrangerá todo o programa. O aluno não terá direito de fazer Segunda Chamada para substituir as notas das listas de exercícios, atividades em sala ou do projeto.

- A verificação do rendimento escolar compreende: frequência e aproveitamento nos estudos, os quais devem ser atingidos conjuntamente;
- É exigida a frequência mínima (FS) 75% (setenta e cinco por cento) das atividades;
- O aluno com frequência suficiente (FS), e média das notas de avaliações do semestre entre 3,0 (três) e 7,0 (sete vírgula zero) terá direito à prova final no semestre;



- Será atribuída nota 0 (zero) ao aluno que não comparecer às atividades ou não apresentar trabalhos no prazo estabelecido;
- Todas as avaliações serão expressas em notas graduadas de 0 (zero) a 10 (dez); e,
- A nota mínima de aprovação na disciplina é 5,0 (cinco vírgula zero). A AVALIAÇÃO FINAL, abrangerá todo o conteúdo do programa.

14. APROVAÇÃO:

Aprovado em ____/____/____

Homologado em ____/____/____

COORDENADOR

GERÊNCIA ACADÊMICA

OBS: As datas das avaliações poderão sofrer alterações de acordo com o disciplinado pela secretaria acadêmica da FASETE.

