

**FACULDADE:** CENTRO UNIVERSITÁRIO DE BRASÍLIA – UniCEUB  
**CURSO:** CIÊNCIA DA COMPUTAÇÃO  
**DISCIPLINA:** CONSTRUÇÃO DE COMPILADORES  
**CARGA HORÁRIA:** 75 H. A. **ANO/SEMESTRE:** 2017/02  
**PROFESSOR:** EDUARDO FERREIRA DOS SANTOS  
**HORÁRIOS:** Quartas às 09h40 e Quintas às 07h40

## PROJETO – CONSTRUÇÃO DE COMPILADORES

### RESUMO

Linguagens de programação são notações para se descrever computações para pessoas e para máquinas. Contudo, apesar de possuírem definições e regras de escrita, tais linguagens ainda precisam ser traduzidas de forma que possam ser executadas por um computador. Os sistemas de software que traduzem essas linguagens são chamados de compiladores. O projeto final da disciplina visa o desenvolvimento de um projeto para construção de um compilador básico de qualquer linguagem para qualquer arquitetura de processadores. A implementação proposta deve ser formalmente provada e minimamente funcional para os propósitos estabelecidos.

### OBJETIVOS

#### Objetivo Geral

Compreender o funcionamento e possibilidades de um compilador.

#### Objetivos Específicos

1. Construir um compilador para qualquer linguagem;
2. Desenvolver um modelo formal para o compilador proposto;
3. Organizar a execução de um projeto de software.

### ORGANIZAÇÃO DO TRABALHO

#### Grupos

- O trabalho poderá ser realizado em grupos de até **3 alunos**, a serem definidos até o dia **17/08/2017**.
- Apesar do trabalho ser em grupo, a produção deve ser individual. Assim, deve ficar **claro e explícito** em todas as etapas do trabalho a contribuição individual de cada um.

### METODOLOGIA

#### Requisitos

Todos os fontes e artefatos produzidos para a disciplina devem ser disponibilizados em um repositório de acesso público, sem a necessidade de cadastro e/ou senha para baixar os fontes.

## METODOLOGIA

Alguns exemplos de repositório público, a saber:

- <https://github.com/>
- <http://bazaar.canonical.com/en/>
- <https://bitbucket.org/>

### **Análise léxica**

Os alunos devem se organizar em grupos e definir as regras para análise léxica a serem implementadas no compilador.

- As regras de análise léxica devem ser representadas em uma representação formal;
- O modelo de análise léxica não precisa ser completo; contudo, precisa estar minimamente definido para ser evoluído ao longo do semestre.

### **Análise sintática e semântica**

Os alunos devem se organizar em grupos e definir as regras para análise sintática a serem implementadas no compilador.

- As regras de análise sintática devem ser representadas em uma representação formal;
- O modelo de análise sintática não precisa ser completo; contudo, precisa estar minimamente definido para ser evoluído ao longo do semestre.

### **Compilador**

Entrega final do compilador. Devem estar inclusos os seguintes itens:

- Modelo de análise léxica;
- Modelo de análise sintática e semântica;
- Gerador de código intermediário;
- Código-fonte do compilador;
- Binários executáveis;
- Documentação mínima de utilização
  - A documentação deve ser entregue em um arquivo de texto junto ao fonte do projeto;
  - Devem fazer parte da documentação todos os requisitos de instalação, execução e compilação do programa;
  - Também devem ser fornecidas na documentação todas as restrições do compilador, ou seja, para qual ambiente/linguagem ele se aplica.
- O projeto será apresentado em um seminário curto de no máximo **15 minutos**.

Observações de ordem geral:

- Os prazos para entrega não serão estendidos, então fiquem atentos às datas;
- Apesar de não haver restrição de linguagem, toda escolha tecnológica deve ser justificada durante o desenvolvimento do trabalho;
- Durante a apresentação cada grupo deve identificar a contribuição individual de cada componente no desenvolvimento do trabalho;
- Caso não seja possível identificar qual foi a contribuição individual de cada aluno, todos terão a menção SR atribuída ao trabalho;
- Após a apresentação será disponibilizado um espaço de 15min para perguntas do professor

## METODOLOGIA

e dos colegas.

## CALENDÁRIO

O trabalho será realizado em etapas, todas constantes do processo de avaliação para a disciplina.

- **Análise léxica: 14/09/2017**
  - Entregar o modelo de análise léxica do compilador.
- **Análise sintática: 26/10/2017**
  - Entregar o modelo de análise sintática do compilador.
- **Compilador: 23/11/2017**
  - Entrega de todos os fontes e/ou modelos utilizados no trabalho.
  - Apresentação de 15min sobre o projeto realizado.

## RECURSOS DIDÁTICOS

### Comunicação entre aluno e professor

- E-mail: [eduardo@eduardosan.com](mailto:eduardo@eduardosan.com)
- Whatsapp
- Portal do professor: <http://www.eduardosan.com/compiladores>
- O espaço aluno será utilizado para comunicar informações sobre: datas das avaliações, plano de ensino, menção do aluno, faltas do aluno e possíveis ausências ou atrasos do professor.
- O portal do professor será utilizado para comunicar informações sobre: plano de ensino, datas das avaliações, lista de exercícios, módulos de ensino (ME), trabalhos, aplicativos e materiais de ensino em geral.

## BIBLIOGRAFIA

### BÁSICA

AHO, Alfred V. E Outros. Compiladores: princípios, técnicas e ferramentas. PEARSON, 2007.  
PRICE, Ana Maria de Alencar. Implementação de linguagens de programação: compiladores. SAGRA-LUZZATTO, 2005.  
RICARTE, Ivan. Introdução à compilação. Elsevier, 2008.

### COMPLEMENTAR

SETHI, R. Compilers: Principles, techniques and tools. Addison-Wesley  
COOPER, K. Engineering a Compiler. Morgan Kaufmann  
APPEL, A.w. Modern Compiler Implementation in Java. Cambridge University Press