

# Guido Araújo

[Início](#)

## MC910 Construção de Compiladores (2s11)

### Notícias

2/8 [Seja bem-vindo!](#)

### Ementa

Esta disciplina cobrirá técnicas, métodos e algoritmos utilizados no projeto de compiladores modernos: análise léxica, sintática e semântica, verificação de tipos, registros de ativação, representação intermediária, geração de código, análise de fluxo de dados, técnicas de otimização, gerenciamento de memória e compilação para linguagens orientadas a objetos.

### Horários

Aulas: Ter. e Qui.

Sala: 361 (IC3.5)

Horários: 10:00 - 12:00

Instrutores: Profs. Guido Araújo e Sandro Rigo

Atendimento: Após a aula.

### Bibliografia

O curso será fortemente baseado no livro-texto abaixo. Os outros livros são auxiliares e podem ajudar na compreensão da matéria.

Modern Compiler Implementation in Java [**livro-texto**], Andrew Appel (é fundamental a utilização da última edição deste livro. Consulte sempre que possível a errata).

Compilers: Principles Techniques and Tools, Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman.

Engineering a Compiler, Keith Cooper.

### Slides

As aulas serão baseadas em um conjunto de slides, disponíveis em anexo abaixo nesta página. Se necessário, notas de aula adicionais, bem como artigos discutidos em classe serão disponibilizados.

### Vídeos

Algumas aulas desta disciplinas estão disponíveis em vídeo. Salientamos que estes vídeos somente poderão ser acessados por alunos regularmente matriculados na disciplina, e que o material é propriedade intelectual do professor e da UNICAMP, não podendo ser distribuído sem a autorização destes.

### Newsgroup

O newsgroup desta disciplina pode ser encontrado [aqui](#).

### Exames

A avaliação será baseada em dois exames com os seguintes pesos: Exame I (50%) e Exame II (50%). Fraudes nos exames não serão toleradas e será atribuída média Zero a todos os envolvidos. Pedidos de revisão de prova devem ser feitos através de e-mail (assunto MC910), até 48 horas após a divulgação do resultado do exame. Alunos com frequência inferior a 75% reprovam-se por falta.

As notas estarão disponíveis nesta Tabela, duas semanas após a realização do Exame.

### Exercícios

Consideramos os exercícios do livro-texto como fundamentais para o bom aprendizado da matéria e a realização dos exames. Recomenda-se fortemente que sejam resolvidos todos os problemas livro-texto.

### Cronograma

2/8 Introdução e tokens

4/8 Expressões regulares

9/8 Autômato Finito (DFA)  
11/8 Conversão NFA para DFA  
16/8 Geradores de analisadores léxicos  
18/8 Analizador descendente LL(1)  
23/8 Eliminação de ambiguidade em LL(1)  
25/8 Analizador LR(0)  
30/8 First, Follow e nullable  
1/9 First, Follow e nullable (cont.)  
6/9 Analizador SLR  
8/9 Analizador LR(1) , LALR(1) e LR(k)  
13/9 Geradores de analisadores sintáticos e diretivas  
15/9 Recuperação de erros  
20/9 Representação intermediária  
22/9 Análise semântica  
27/9 Registro de ativação  
29/9 Registro de ativação (cont.)  
**4/10 Exame I**  
**6/10 Não haverá aula**  
11/10 Blocos Básicos  
13/10 Seleção de instruções  
18/10 Análise de fluxo de dados  
20/10 Otimização de código  
25/10 Otimização de código  
**27/10 Não haverá aula**  
1/11 Análise de longevidade  
3/11 Alocação de registradores  
8/11 Alocação de registradores  
10/11 Garbage collection  
**15/11 Feriado**  
17/11 Garbage collection  
22/11 Compilação para linguagens OO  
24/11 Técnicas avançadas de compilação  
**29/11 Exame II**  
**13/12 Exame Final**

Anexo	Tamanho
<a href="#">cap2-lex.pdf</a>	1.25 MB
<a href="#">cap3-parser.pdf</a>	726.03 KB
<a href="#">cap3-LR-parser.pdf</a>	1.53 MB
<a href="#">cap4-IR.pdf</a>	500.65 KB
<a href="#">cap5-semant.pdf</a>	352.36 KB
<a href="#">cap6-frames.pdf</a>	268.33 KB