



# Compiladores

3º Ano do MIEIC

João M. P. Cardoso



Dep. de Engenharia Informática  
Faculdade de Engenharia (FEUP)  
Universidade do Porto  
Porto  
Portugal  
Email: [jmpc@acm.org](mailto:jmpc@acm.org)

# Docentes

## ➤ 2013/2014

- João M. P. Cardoso
  - Gab. I337
  - Horário de Atendimento: ver moodle
  - E-mail: [jmpc@fe.up.pt](mailto:jmpc@fe.up.pt)
- Rui Maranhão
  - Gab. I226
  - Horário de Atendimento: ver moodle
  - E-mail: [rma@fe.up.pt](mailto:rma@fe.up.pt)

# Páginas da Disciplina

- SiFEUP (rules, timetable, list of students, etc.):
  - [https://sigarra.up.pt/feup/pt/ucurr\\_geral.ficha\\_uc\\_view?pv\\_ocorrencia\\_id=333128](https://sigarra.up.pt/feup/pt/ucurr_geral.ficha_uc_view?pv_ocorrencia_id=333128)
- Moodle (organization of the course, mailing-lists, etc.):
  - <http://moodle.up.pt/course/view.php?id=699>
- Google Drive (files, documents, etc.):
  - <https://drive.google.com/?tab=wo&authuser=0#folders/0ByA8BQ9Y5E9OMjBYY2p3REZXRWc>

# Objectivos, Competências e Resultados de aprendizagem

- Fornecer os conceitos que permitam:
  - compreender as fases de compilação de linguagens, em especial das linguagens imperativas e orientada por objectos;
  - 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

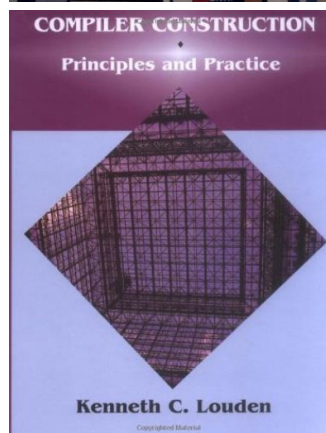
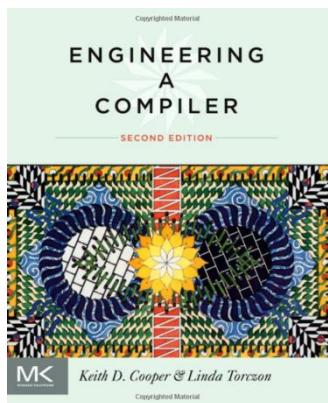
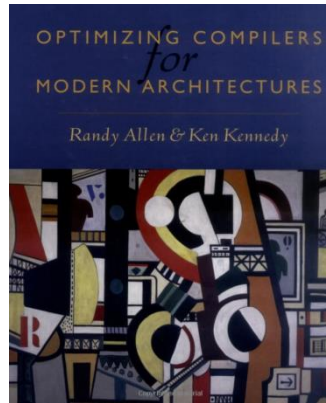
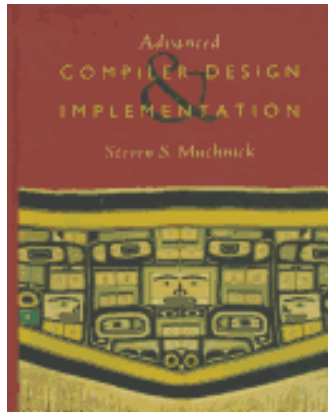
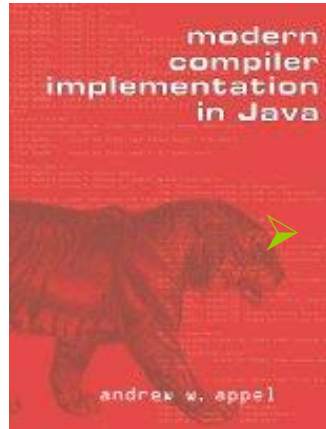
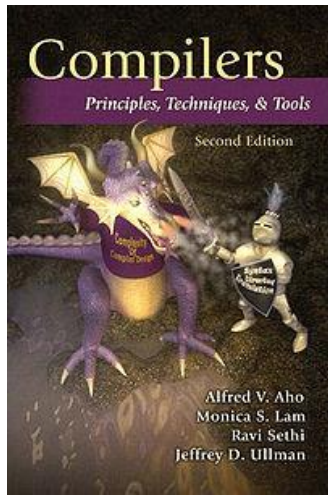
# Objectivos, Competências e Resultados de aprendizagem

- As competências adquiridas permitirão aos estudantes:
  - desenvolver e implementar em software processadores de linguagens artificiais e de informação especificada textualmente segundo determinadas regras lexicais e sintácticas;
  - conceber e implementar em software as várias etapas relacionadas com compiladores, nomeadamente:
    - expressões regulares e autómatos finitos;
    - analisadores sintácticos e semânticos;
    - analisadores semânticos;
    - optimizações de código;
    - geradores de código para processadores ou para máquinas virtuais;

# Programa

- Introdução. Fases da compilação e estrutura típica de um compilador.
- Análise lexical. Expressões regulares e autómatos finitos.
- Análise sintáctica. Gramáticas. Implementação de analisadores sintácticos. Tratamento de erros.
- Análise semântica. Implementação de analisadores semânticos. Verificação de tipos.
- Ambientes de execução. Organização de memória e mecanismos para a passagem de parâmetros.
- Geração de código intermédio de baixo e de alto-nível.
- Técnicas de geração de código final e de optimização.





# Bibliografia

## Bibliografia Principal

- A. Aho, M. Lam, R. Sethi, J. Ullman, *Compilers: Principles, Techniques, and Tools*, 2nd Edition, Addison Wesley, 2007. ISBN: 0321486811 (Existe 1ª edição (1986) na biblioteca)
- [Appel, Andrew Wilson](#), *Modern Compiler Implementation in Java*, 2nd edition. Cambridge University Press, 2002. [ISBN 0-521-82060-X](#)

## Bibliografia Complementar

- [Muchnick, Steven](#), *Advanced Compiler Design and Implementation*, Morgan Kaufman Publishers, 1997. [ISBN 1-55860-320-4](#)
- Allen, Randy; and [Kennedy, Ken](#), *Optimizing Compilers for Modern Architectures*, Morgan Kaufman Publishers, 2001. [ISBN 1-55860-286-0](#)
- Cooper, Keith D., and Torczon, Linda, [Engineering a Compiler](#), Morgan Kaufmann, 2nd edition, February 21, 2011. ISBN 10: 012088478X
- Loudon, Kenneth C.; [Compiler construction](#). Course Technology, ISBN 0-534-93972-4

# Métodos de Ensino

## ➤ Aulas teóricas:

- exposição dos assuntos, acompanhada por exemplos e pela apresentação de pistas para resolução dos problemas das aulas práticas e do trabalho prático

## ➤ Aulas teórico-práticas:

- resolução e discussão de questões relacionadas com o trabalho prático



# Avaliação

## ➤ **Modo de Avaliação**

- Avaliação distribuída sem exame final

## ➤ **Obtenção de Frequência**

- Realização do trabalho prático com aproveitamento (nota igual ou superior a 50%).

## ➤ **Cálculo da Classificação Final**

- ÉPOCA NORMAL:  
Nota final =  $\text{ROUND}(0,70 \cdot \text{AD} + 0,15 \cdot \text{T1} + 0,15 \cdot \text{T2})$ 
  - AD: nota da avaliação distribuída (trabalho prático) [0..20]
  - T1: nota no primeiro teste [0..20]
  - T2: nota no segundo teste [0..20]O aluno obtém aproveitamento na disciplina na época normal se obtiver frequência e se obtiver uma nota média nos testes (T1 e T2) igual ou superior a 8 valores.
- ÉPOCA DE RECURSO:  
Nota final =  $\text{ROUND}(0,70 \cdot \text{AD} + 0,30 \cdot \text{EX})$ 
  - AD: nota da avaliação distribuída (trabalho prático) [0..20]
  - EX: nota no exame de recurso [0..20]O aluno obtém aproveitamento na disciplina na época de recurso se obtiver frequência e se obtiver uma nota de exame (EX) igual ou superior a 8 valores.

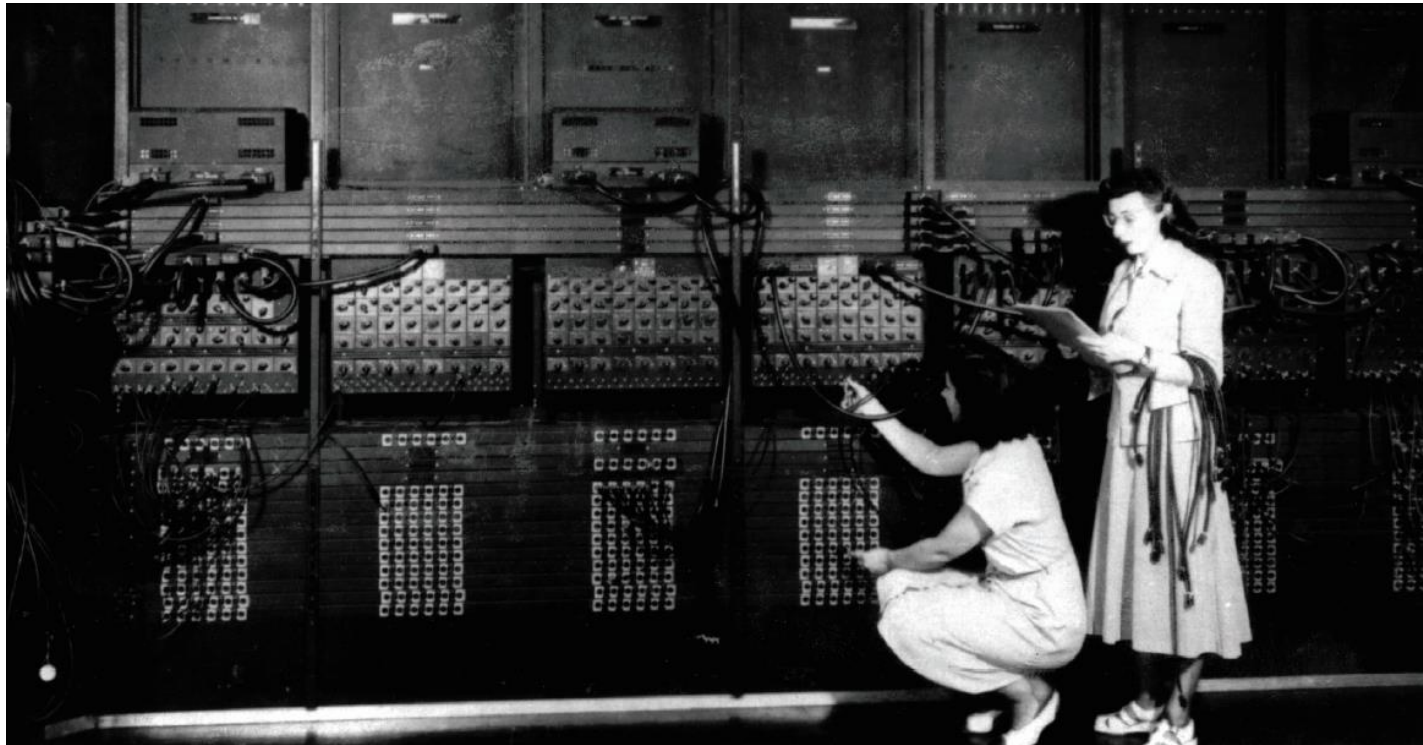
## ➤ **AD: nota da avaliação distribuída (trabalho prático) [0..20]**

- Participação e assiduidade: 10%
- Trabalho:
  - Primeiro checkpoint: 10%
  - Segundo checkpoint: 10%
  - Entrega Final: 50%
- Apresentação/Discussão: 20%

# Software

- JavaCC, <https://javacc.dev.java.net/>
- JASMIN, <http://jasmin.sourceforge.net/>
- COINS, <http://www.coins-project.org/international/>
- ANTLR - Another Tool for Language Recognition, <http://wwwantlr.org/>
- IDE:
  - Eclipse, NetBeans

- Two programmers wiring the right side of the ENIAC with a new program



Source:

**Actually, Turing Did Not Invent the Computer**

By Thomas Haigh

Communications of the ACM, Vol. 57 No. 1, 2014, pp. 36-41



# BOM TRABALHO!