



## COMPILAÇÃO | 21018

### Data e hora de realização

26 de julho de 2025 às 15:00 de Portugal Continental

### Duração da prova

90 minutos + 60 minutos de tolerância

### Conteúdos

Compiladores. Fases da compilação. Análise léxica. Análise sintática. Geração de código. Otimização de código.

### Competências

Deve demonstrar capacidades para:

1. Distinguir as várias fases de um compilador e compreender cada uma delas no contexto global de tradução de uma linguagem para outra.
2. Desenvolver um compilador para uma dada linguagem, tendo em conta as diversas fases e as ferramentas disponíveis.

### Critérios de avaliação e cotação

Cotações:

Análise léxica + Análise sintática – 3 valores

Geração e otimização de código intermédio – 3 valores

Geração de código final – 2 valores

Questão individual – 4 valores (1 valor por alínea) ´

### **Normas a respeitar**

Deve submeter a sua prova na plataforma WISEflow em formato PDF (tamanho máximo: 20MB), identificando-se no início (nome, turma e número de estudante).

Caso resolva parte do exame em papel, deve garantir a qualidade e legibilidade da imagem digitalizada, dado que esta terá de ser inserida na Folha de Resolução. Também deverá ter em atenção o tamanho total do ficheiro, dada a limitação da plataforma (20 MB).

Deve carregar o referido ficheiro para a plataforma WISEflow até à data e hora limite de entrega. Evite a entrega próximo da hora limite para se precaver contra eventuais problemas.

No mesmo prazo, deve colocar em anexo o ficheiro ZIP, de acordo com as indicações dadas.

Votos de bom trabalho!

Rudi Gualter e Constantino Martins

## **Trabalho a desenvolver**

O trabalho a ser entregue corresponde a duas partes: entrega da versão final do trabalho em grupo desenvolvido ao longo do semestre e resolução de uma questão individual a resolver durante o horário da prova.

O trabalho em grupo é a implementação de um compilador para a linguagem MOCC, que temos trabalhado durante o semestre.

Nesta fase deve apresentar o compilador completo, com todas as fases relativas aos e-fólios A e B, com todas as melhorias que tenham sido feitas, entretanto, e ainda a geração de código final. Caso a versão final não esteja funcional, deve enviar na mesma esta versão e também última versão funcional, indicando as diferenças entre ambas.

A linguagem para o código final fica ao vosso critério. Pode ser uma linguagem assembly, existem simuladores online disponíveis para vos ajudar, inclusive podem usar P3, já aprendido no curso, ou ainda bytecode de Java. Assim, não se esqueçam de identificar a linguagem e um simulador para testar o código final.

Além disso, devem enviar um relatório sobre o trabalho, indicando as diferenças relativamente ao enviado no e-fólio B.

**NOTA MUITO IMPORTANTE:** Todos os membros de cada grupo devem, individualmente, submeter o e-fólio global no local de entrega durante o horário em que decorre a prova. Quem não o fizer não poderá ser classificado, mesmo que todos os outros colegas do grupo o tenham feito.

Assim, deve ser entregue o ficheiro PDF com a questão inicial no local designado, e um ficheiro com a extensão ZIP em anexo, incluindo:

- todo o código usado na construção do compilador;
- um ficheiro README.TXT com as instruções para compilação do compilador, incluindo versões do software utilizado;
- um ficheiro executável funcional (para Windows ou Linux) do compilador que permita executar e testar o código caso não seja possível compilar o mesmo;
- um relatório de grupo, conforme a explicação acima.

### **Questão Individual**

Considere a linguagem definida pela seguinte gramática:

Prog --> Inst Prog | 'fim'

Inst --> Atrib | IfCond

Atrib --> ID '=' Expr ';'

IfCond -> 'if' '(' Cond ')' Inst

Cond -> Expr RelOp Expr

RelOp -> '<' | '<=' | '>' | '>=' | '==' | '!='

Expr --> Expr '+' Term | Term

Term --> Term '\*' Factor | Factor

Factor --> '(' Expr ')' | ID | NUMBER

E considere o seguinte código nessa mesma linguagem:

a = 5 ;

b = 2 ;

if (a > b) b = b + 1 ;

fim

Em cada alínea, indique os passos que seriam dados por um compilador (sem precisar de implementar):

- a) Tabela com valores e *tokens* respetivos após análise léxica;
- b) Derivação da gramática para o exemplo dado.
- c) Código intermédio gerado.
- d) Código intermédio otimizado, com indicação das otimizações realizadas.

**FIM**