



Distribuição semanal das aulas		
Terça	10h00–12h00	Aula teórica
Quinta	10h00–12h00	Aula teórica
Sexta	10h00–12h00	Laboratório (turmas AB)
Terça	16h00–17h00	Laboratório (turmas CD)

PROF. ALEXANDRE FALCÃO  
 afalcao@ic.unicamp.br

Informações específicas para as turmas ABCD estão disponíveis em:

<http://www.ic.unicamp.br/~afalcao/mc102>

Informações gerais sobre o oferecimento da disciplina MC102 estão disponíveis em:

<http://www.ic.unicamp.br/~mc102>

**Ementa:** Conceitos básicos de organização de computadores. Construção de algoritmos e sua representação em pseudocódigo e linguagens de alto nível. Desenvolvimento sistemático e implementação de programas. Estruturação, depuração, testes e documentação de programas. Resolução de problemas.

**Linguagem de programação:** Python, versão 3.

**Critério de Avaliação:** Serão levados em consideração os seguintes elementos avaliativos:

- a) **Atividades Conceituais:** são atividades elaboradas com o intuito de avaliar o grau de compreensão apresentado pelos(as) estudantes dos conceitos referentes a determinado(s) tópico(s) da disciplina de maneira a auxiliá-los(as) a regular seu processo de aprendizagem. A nota atribuída a cada uma destas atividades será uma nota de participação, feita de maneira proporcional ao número de questões respondidas pelo(a) aluno(a), independentemente do fato de estarem corretas ou não.

Para a realização das atividades serão disponibilizados questionários de múltipla escolha na área **MC102—2S2018—Algoritmos e Programação de Computadores** do sistema Moodle, acessível a partir do endereço web <http://www.ggte.unicamp.br/eam>.

Durante o semestre serão propostas  $n$  Atividades Conceituais, com média  $M_{AC}$ .

- b) **Tarefas de laboratório:** serão propostas  $m$  tarefas de laboratório, que deverão ser implementadas pelos(as) alunos(as) de maneira individual. Os programas desenvolvidos serão testados com um jogo de testes pré-determinado, subdividido em *testes abertos*, que podem ser vistos pelos(as) alunos(as) e *testes fechados*, que não podem ser vistos. A nota de cada laboratório será proporcional ao número de testes, abertos ou fechados, que executaram corretamente. No entanto, serão considerados válidos apenas os programas projetados de maneira a resolver o problema proposto para um conjunto amplo de possibilidades e não aqueles projetados para emitir a saída correta para o subconjunto de testes abertos.

O gerenciamento da submissão e testes das tarefas de laboratório será feito por um sistema automatizado conhecido como *SuSy*, acessível a partir do endereço web <https://susy.ic.unicamp.br:9999/mc102geral>.

Juntamente com o enunciado de cada tarefa de laboratório  $L_i$  será indicado o peso desta tarefa  $LP_i \in \{1, 2, 3\}$ . A média dos laboratórios,  $M_L$ , é a média ponderada destas  $m$  notas.

- c) **Provas teóricas:** Serão realizadas duas provas teóricas,  $P_1$  e  $P_2$ , em papel, de forma individual e sem nenhum tipo de consulta nas seguintes datas e com os seguintes pesos:

Prova	Peso	Data	Horário
$P_1$	2	20 de setembro	10h00–12h00
$P_2$	3	27 de novembro	10h00–12h00

A média de provas teóricas,  $M_P$ , é a média ponderada das provas  $P_1$  e  $P_2$  considerando seus respectivos pesos.

**Frequência mínima:** De acordo com o Regimento Geral de Graduação os(as) alunos(as) devem ter frequência *Freq* maior ou igual a 75% para aprovação.

**Resultado final:** A média final  $M_{Final}$  e a situação do(a) aluno(a) serão definidas de acordo com as regras abaixo:

- Caso  $Freq \geq 75\%$ :
  - Se  $M_P \geq 5.0$  e  $M_L \geq 5.0$ :
    - \* Será calculada uma média ponderada dos elementos avaliativos:
$$M_{Elem} = (0.6 * M_P) + (0.3 * M_L) + (0.1 * M_{AC})$$
    - \* O(A) aluno(a) estará *aprovado(a) por nota e frequência* com média final  $M_{Final} = \text{Max}(5, M_{Elem})$
  - Se  $M_P < 2.5$  ou  $M_L < 2.5$ :
    - \* O(A) aluno(a) estará *reprovado(a) por nota*, com média final  $M_{Final} = \text{Min}(M_P, M_L)$ .
  - Senão:
    - \* O(A) aluno(a) deverá fazer o exame  $E$ , a ser realizado no dia 11 de dezembro, das 10h00–12h00.
    - \* Será calculada a média pré-exame:  $M_{Pre} = \text{Min}(M_P, M_L)$
    - \* Sua média final será  $M_{Final} = \frac{M_{Pre} + E}{2}$ .
    - \* Caso o(a) aluno(a) tenha obtido  $M_{Final} \geq 5.0$  estará *aprovado(a) por nota e frequência*, caso contrário estará *reprovado(a) por nota*.
- Caso  $Freq < 75\%$ :
  - O(A) aluno(a) estará *reprovado(a) por frequência*, com média final  $M_{Final} = \text{Min}(M_P, M_L)$ .

**Atendimento:** Este oferecimento de MC102 conta com a colaboração de vários alunos de pós-graduação (PEDs) e graduação (PADs) que atuam como monitores(as). Além do atendimento durante as aulas de laboratório, haverá plantões de dúvidas presenciais e aulas extras de revisão em horários e salas que serão divulgados na página web geral da disciplina.

### Informações Adicionais:

1. Nos dias de prova será necessária a apresentação de documento oficial com foto.
2. Não há possibilidade de troca de horário de provas e/ou exame final.
3. Não haverá reposição de tarefas de laboratório.
4. Todas as tarefas são individuais. Qualquer tentativa de fraude implicará em nota *0.0 (zero)* na disciplina para **todas** as pessoas envolvidas.
5. Fraudes poderão ser detectadas automaticamente entre *todas* as respostas (de *todos* os alunos e de *todas* as alunas, entre *todas* as turmas) ao longo do semestre.
6. A submissão de um código que não produz as saídas corretas dos testes via a implementação dos algoritmos solicitados nas tarefas de laboratório, mas que exhibe as saídas esperadas dos testes abertos a partir da comparação de trechos da entrada será considerada fraude.

**Material de Apoio:** recomendamos as páginas oficiais da linguagem Python: [www.python.org](http://www.python.org) ou [www.python.org.br](http://www.python.org.br) (em português). Indicamos também o livro *How to Think Like a Computer Scientist: Interactive Edition* de Brad Miller e David Ranum, disponível em inglês e português, conforme descrito a seguir:

- How to Think Like a Computer Scientist: Interactive Edition. Disponível em: <https://runestone.academy/runestone/static/thinkcspy/index.html>
- Aprendendo com Python: Edição interativa (usando Python 3.x) Disponível em: <https://panda.ime.usp.br/pensepy/static/pensepy/index.html>

Outros textos e alguns vídeos estão indicados na página web geral da disciplina. Caso consulte algum material, verifique se a versão utilizada de Python é a 3. Se não for, fique atento(a) aos detalhes que variam de uma versão para outra.