



PROF. MARCIO MACHADO PEREIRA
mpereira@ic.unicamp.br

Distribuição semanal das aulas por tipo		
Tipo	Dia da semana	Horário
Teórica (turmas GHIJ)	Terça	10h00–12h00
Teórica (turmas GHIJ)	Quinta	10h00–12h00
Laboratório (turmas GH)	Quinta	14h00–16h00
Laboratório (turmas IJ)	Quarta	16h00–18h00

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

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

Informações específicas das turmas GHIJ estão disponíveis em:

<https://iviarcio.wordpress.com>

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: Neste semestre será adotada a linguagem Python, versão 3.

Critério de Avaliação: Serão levados em consideração os seguintes elementos: a) Atividades Conceituais, b) Tarefas de Laboratório e c) Provas Teóricas.

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. Para que possam cumprir efetivamente seu papel, estas atividades deverão ser realizadas de maneira individual e sem nenhum tipo de consulta. As Atividades Conceituais poderão ser aplicadas ao longo do semestre de duas formas:

- Questionários Online: serão disponibilizados questionários de múltipla escolha no sistema Moodle acessível a partir do endereço web <http://www.ggte.unicamp.br/eam> na área **MC102 – 1S2018 - Algoritmos e Programação de Computadores**.
- Tarefas Presenciais: serão atividades também disponibilizadas via Moodle em área específica da turma do(a) aluno(a) e realizadas, sob supervisão, durante algumas aulas de laboratório. As datas destas atividades serão divulgadas na página web geral da disciplina com pelo menos duas semanas de antecedência.

Durante o semestre serão propostas n Atividades Conceituais. A nota atribuída a cada uma destas atividades AC_i será feita de maneira proporcional ao número de questões respondidas pelo aluno. Os Questionários Online terão peso 1 e as Tarefas Presenciais peso 3. A média das Atividades Conceituais, M_{AC} , é a média ponderada das n notas de Atividades Conceituais respeitando seus respectivos pesos.

b) **Tarefas de laboratório:** Haverá m tarefas de laboratório, nas quais os(as) alunos(as) deverão implementar problemas propostos de maneira individual. O gerenciamento da submissão e testes das tarefas de laboratório será feito por um sistema automatizado conhecido como *SuSy*, acessível no endereço web <https://susy.ic.unicamp.br:9999/mc102geral>.

Os programas submetidos 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. Juntamente com o enunciado de cada tarefa de laboratório L_i serão indicados o peso desta tarefa $LP_i \in \{1, 2, 3\}$, as regras específicas para a sua implementação e também a fórmula que indicará a nota das soluções válidas em função do número de testes abertos ou fechados que retornaram resultados corretos. A média dos laboratórios, M_L , é a média ponderada das m notas de laboratórios L_i respeitando seus pesos LP_i .

- 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	19 de abril	10h00–12h00
P_2	3	21 de junho	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 sua presença às aulas controlada e a frequência $Freq$ deve ser maior ou igual a 75% para aprovação.

Resultado final: Seja M_{Elem} a média ponderada dos elementos da avaliação seguindo a fórmula:

$$M_{Elem} = (0.6 * M_P) + (0.3 * M_L) + (0.1 * M_{AC})$$

e $M = \min(M_{Elem}, M_P, M_L)$ a média sem exame. A média final F e a situação do(a) aluno(a) serão definidas de acordo com as regras abaixo.

- Caso $Freq \geq 75\%$:
 - Caso $M \geq 5$: o(a) aluno(a) estará *aprovado(a) por nota e frequência* com $F = M$
 - Caso $2.5 \leq M < 5$: o(a) aluno(a) terá direito a fazer o exame E no dia 12 de julho das 10h00–12h00. Sua média final será $F = \frac{M + E}{2}$. Caso o(a) aluno(a) tenha obtido $F \geq 5.0$ estará *aprovado(a) por nota e frequência*, caso contrário estará *reprovado(a) por nota*.
 - Caso $M < 2.5$ o(a) aluno(a) estará *reprovado(a) por nota* com $F = M$.
- Caso $Freq < 75\%$ o(a) aluno(a) estará *reprovado(a) por frequência* com $F = M$.

Atendimento: Este oferecimento de MC102 conta com a colaboração de vários(as) alunos(as) 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.
5. Qualquer tentativa de fraude implicará em nota 0.0 (*zero*) na disciplina para **todas** as pessoas envolvidas.
6. O sistema de correção automático dispõe de um eficiente detector de fraudes. Ele é programado para detectar plágios entre *todas* as respostas (de *todos* os alunos e de *todas* as alunas, entre *todas* as turmas) ao longo do semestre. Ocorrência de fraude implicará em nota 0.0 (*zero*) na disciplina para **todas** as pessoas envolvidas.

Material de Apoio: Existem muitos textos sobre programação em Python. Neste semestre utilizaremos como referência principal 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>

Caso tenha interesse por outro texto, verifique se a versão utilizada de Python é a 3. Se não for, fique atento aos detalhes que variam de uma versão para outra.

O material teórico do curso está disponível em <http://github.com/iviarcio/mc102>. Adicionalmente, recomendamos o material disponível nas páginas oficiais da linguagem Python: www.python.org ou www.python.org.br (em português).