

# MC102 - Algoritmos e Programação de Computadores

Turmas Coordenadas - Segundo Semestre de 2022

## Conteúdo desta página

- [Avisos Importantes](#)
- [Turmas e Docentes](#)
- [Formato das Aulas](#)
- [Material Didático](#)
- [Listas de Exercícios](#)
- [Atividades Práticas](#)
- [Sites Recomendados](#)
- [Downloads](#)
- [Ementa da Disciplina](#)
- [Programa da Disciplina](#)
- [Avaliação](#)
- [Datas Importantes](#)

## Avisos Importantes

- [01/08/2022] Divulgado o [calendário da disciplina](#).

## Turmas e Docentes

- Turmas ABCD:
  - Professor [Víctor Eduardo Martínez Abaunza](#)
  - Email: [vemartin@unicamp.br](mailto:vemartin@unicamp.br)
- Turmas EF:
  - Professor [Zanoni Dias](#)
  - Email: [zanoni@ic.unicamp.br](mailto:zanoni@ic.unicamp.br)
- Turmas IJKL:
  - Professor [Guilherme Pimentel Telles](#)
  - Email: [gpt@ic.unicamp.br](mailto:gpt@ic.unicamp.br)
- Turmas MNOP:
  - Professor [Paulo Lício de Geus](#)
  - Email: [paulo@ic.unicamp.br](mailto:paulo@ic.unicamp.br)
- Turmas RS:
  - Professor [Samuel Praça de Paula](#)
  - Email: [s233554@dac.unicamp.br](mailto:s233554@dac.unicamp.br)
- Turmas UVX:
  - Professor [Julio César López Hernández](#)
  - Email: [jlopez@ic.unicamp.br](mailto:jlopez@ic.unicamp.br)
- Turmas Z:
  - Professor [Ricardo Dahab](#)
  - Email: [rdahab@ic.unicamp.br](mailto:rdahab@ic.unicamp.br)

### Observações:

- Toda mensagem enviada para os professores deve conter no começo do subject/assunto o texto “[MC102]” seguido do motivo da mensagem.  
Exemplo: [MC102] Problema com o Sistema de Submissão
- O contato por email só deve ser usado em casos excepcionais.
- Para contatos por email, sempre use emails acadêmicos (@dac.unicamp.br) e identifique-se corretamente (nome completo, turma e RA).
- Emails que não seguirem estritamente as regras acima não serão respondidos.
- Dúvidas quanto ao conteúdo das aulas ou às atividades práticas não serão respondidas por email. Nesses casos, consulte o [Plantão Online de Dúvidas](#).

## Material Didático

Materiais didáticos recomendados:

- Aula Introdutória [ [slides](#) ] [ [vídeo](#) ]
- Primeira Aula de Laboratório [ [slides](#) ] [ [vídeo](#) ]
- Python Básico: Tipos, Variáveis, Operadores, Entrada e Saída [ [slides](#) ] [ [vídeo](#) ]
- Comandos Condicionais [ [slides](#) ] [ [vídeo](#) ]
- Comandos de Repetição [ [slides](#) ] [ [vídeo](#) ]
- Listas e Tuplas [ [slides](#) ] [ [vídeo](#) ]
- Strings [ [slides](#) ] [ [vídeo](#) ]
- Dicionários [ [slides](#) ] [ [vídeo](#) ]
- Funções [ [slides](#) ] [ [vídeo](#) ]
- Objetos Multidimensionais [ [slides](#) ] [ [vídeo](#) ]
- Algoritmos de Ordenação [ [slides](#) ] [ [vídeo](#) ]
- Algoritmos de Busca [ [slides](#) ] [ [vídeo](#) ]
- Recursão [ [slides](#) ] [ [vídeo](#) ]
- Algoritmos de Ordenação Recursivos [ [slides](#) ] [ [vídeo](#) ]
- Arquivos [ [slides](#) ] [ [vídeo](#) ] (extra)
- Expressões Regulares [ [slides](#) ] [ [vídeo](#) ] (extra)
- Execução de Testes no Google Cloud Shell [ [slides](#) ] [ [vídeo](#) ] (extra)
- Numpy [ [slides](#) ] [ [vídeo](#) ] (extra)
- Pandas [ [slides](#) ] [ [vídeo](#) ] (extra)

Materiais didáticos complementares:

- Panda - Cursos de Computação em Python (IME -USP):
  - [Aulas de Introdução à Computação em Python](#)
  - [Como Pensar Como um Cientista da Computação](#)
- [Material Didático de MC102 \(Marcio Pereira\)](#).

Videoaulas:

- [Introdução à Ciência da Computação com Python - Fabio Kon](#) (em português)
- [Python Programming Tutorials - Socratica](#) (em inglês)

## Listas de Exercícios

- [Primeira Lista - Tipos, Variáveis, Operações Matemáticas e Comandos Condicionais](#)
- [Segunda Lista - Comandos de Repetição](#)
- [Terceira Lista - Lista e Tuplas](#)
- [Quarta Lista - Strings](#)
- [Quinta Lista - Funções](#)
- [Sexta Lista - Recursão](#)

## Atividades Práticas

- As atividades práticas serão disponibilizados no [SuSy](#).

## Sites Recomendados

Os sites listados abaixo fornecem informações adicionais aos tópicos cobertos nas aulas:

- MC102 - Algoritmos e Programação de Computadores:
  - [Site da Coordenação de MC102](#)
- Programação em Python:
  - [Google Cloud Shell](#)
  - [Python Online \(Repl.it\)](#)
  - [The Python Tutorial](#)
  - [The Python Language Reference](#)
  - [Python Programming Examples](#)
  - [Python Tutorial for Beginners](#)
  - [LearnPython.org](#)
- Ambientes de Desenvolvimento Integrado (IDE) para Python:
  - [PyCharm](#)
  - [Visual Studio Code](#)
  - [Jupyter](#)
  - [Atom](#)
  - [Spyder](#)
- Wikipédia (em português):
  - [Computador](#)
  - [Linguagem de Programação C](#)
  - [Ciência da Computação](#)

- [História da Computação](#)
- [Compilador](#) x [Interpretador](#)
- [Tabela ASCII](#)
- [Sistema Binário](#)
- [Linux](#)
- [Comandos para Linux/Unix](#)
- [Emacs / Vim](#)
- Competições de Programação:
  - [Olimpíada Brasileira de Informática](#)
  - [Maratona de Programação](#)
  - [UVA Online Judge - Contest Problem Set Archive](#)
  - [URI Online Judge](#)
  - [Sphere Online Judge \(SPOJ\)](#)
  - [CodeChef](#)
  - [Codeforces](#)

## Downloads

- Reference Cards:
  - [Python](#) (em inglês)
  - [Linux](#) (em inglês)
  - [Emacs](#) (em inglês)
  - [Vim](#) (em português)
- Jogos (*Programming Games*):
  - Cargo-Bot ( [iPad](#) )
  - Robo Logic 2 HD - Lite ( [iPad](#) )
  - Light-Bot ( [iPhone/iPad](#) / [Android](#) )
  - Hopscotch ( [iPhone/iPad](#) )
  - Swift Playgrounds ( [iPad](#) )

## Ementa da Disciplina

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.

## Programa da Disciplina

- Conceitos Básicos de Organização de Computadores
- Conceitos Básicos sobre Algoritmos e Programação
- Entrada e Saída de Dados
- Expressões Aritméticas, Lógicas e Relacionais
- Comandos Condicionais
- Comandos de Repetição
- Listas, Tuplas, Dicionários, Strings e Matrizes
- Funções e Escopo de Variáveis
- Algoritmos de Ordenação
- Algoritmos de Busca
- Recursão
- Algoritmos Recursivos de Ordenação
- Arquivos (extra)
- Expressões Regulares (extra)

## Avaliação

A avaliação será realizada através de tarefas práticas, ou seja, programas em Python que deverão ser implementados e submetidos para correção automática através do [SuSy](#).

Serão propostas  $n$  tarefas práticas, que deverão ser realizadas pelos alunos. Os programas desenvolvidos serão testados com um conjunto pré-determinado de testes, subdividido em testes abertos (que podem ser acessados pelos alunos) e testes fechados (que não podem). A nota de cada atividade prática será proporcional ao número de testes, abertos ou fechados, que executarem corretamente.

Juntamente com o enunciado de cada tarefa prática será indicado o peso  $P_i \in \{1, 2, 3, 4\}$  dessa tarefa. A média das tarefas práticas ( $P$ ) será a média ponderada das notas das tarefas.

A média final  $F$  e a situação de cada aluno serão definidas de acordo com as regras a seguir.

- Caso  $P \geq 5$ :

O aluno estará aprovado com média final ( $F$ ):

$$F = P$$

- Caso  $2,5 \leq P < 5$ :

O aluno poderá realizar o exame. O exame será composto por um subconjunto das tarefas práticas disponibilizadas ao longo do semestre. O aluno poderá refazer as tarefas indicadas no período do exame ou aproveitar os programas já submetidos anteriormente naquelas tarefas (sem necessidade de refazer a tarefa).

A nota do exame ( $E$ ) será calculada como a média ponderada das tarefas selecionadas para compor o exame. Os pesos das tarefas para fins do exame poderão ser diferentes daqueles previamente utilizados para o cálculo da média das tarefas práticas ( $P$ ). O cálculo da média final ( $F$ ) será feita da seguinte forma:

$$F = \min\{5, (P + E)/2\}$$

Caso  $F \geq 5,0$  o aluno estará aprovado. Caso contrário, estará reprovado.

- Caso  $P < 2,5$ :

O aluno estará reprovado, com média final ( $F$ ):

$$F = P$$

#### Observações:

1. Não haverá atividades práticas substitutivas.
2. As tarefas práticas serão disponibilizadas às segundas-feiras de manhã e ficarão disponíveis por no mínimo uma semana e no máximo três semanas (sempre até domingo a noite). Recomenda-se que todas as tarefas sejam realizadas na primeira semana que elas forem disponibilizadas, já que toda semana uma nova tarefa será disponibilizada.
3. Qualquer tentativa de plágio ou fraude nas atividades práticas implicará em nota final  $F = 0$  (zero) para todos os envolvidos, sem prejuízo de outras sanções. Exemplos de plágios e fraudes:
  - Compartilhamento, cópia ou compra de programas.
  - Submissão de programas que produzam as saídas esperadas dos testes abertos a partir da comparação de trechos da entrada, sem de fato implementar os algoritmos solicitados nas tarefas práticas.
4. Casos de tentativa de plágio ou fraude nas tarefas práticas poderão ser detectadas automaticamente entre todas as submissões (de todos os alunos, de todas as turmas) ao longo do semestre.
5. Todos os casos de tentativa de plágio ou fraude automaticamente detectados serão verificados manualmente pela coordenação da disciplina até o final do semestre.
6. As notas das atividades práticas serão divulgadas no [site da disciplina](#) até, no máximo, uma semana após o final do prazo de submissão das tarefas.
7. De acordo com a fórmula acima, caso um aluno seja aprovado após realizar o exame final, sua nota final será  $F = 5$  (cinco).

## Datas Importantes

- [Calendário da disciplina](#): consulte as datas das atividades práticas e os tópicos cobertos a cada semana de aula.
- [Calendário oficial da DAC](#): consulte as datas sem atividades didáticas na Unicamp, além de outras datas importantes como o período de alteração e desistência de disciplinas.
- Exame: conforme indicado no [calendário da disciplina](#), os laboratórios correspondentes ao exame estarão disponíveis para submissão de 15 a 21/12/2022.

