

MC-102 turmas K e L - 2024

Professores: Jacques Wainer wainer@unicamp.br e Rodolfo Jardim rja@unicamp.br

Aulas: 4a das 14 as 16 CB11, 6a das 14 as 16 CB12

Laboratórios: 2a das 14 as 16 SI05 e SI10

Sobre essa turma

Esta turma será diferente das outras turmas de MC102.

Nesta turma **vamos assumir** que você irá usar uma IA generativa para gerar o programa.

IAs generativas:

- ChaptGPT (3.5 - gratuito)
- CoPilot (da GitHub, não o CoPilot da Microsoft) que é gratuito para pessoal ligado a educação
- vc pode usar outras que você queira

No ensino de programação tradicional, o que o aluno aprende é **usar** os comandos e conceitos da linguagem de programação para criar um programa a partir de uma descrição do problema.

Para vocês, a IA generativa vai criar o programa a partir da descrição do problema. O seu papel é

- interagir com a IA para especificar o problema
- **testar** a solução criada pela IA
- entender os comandos usados na solução
- entender em algum nível a solução apresentada pela IA
- aprender a quebrar problemas grandes em partes menores, e pedir que a IA resolva essas partes menores.

ChatGPT não sabe programar. Num certo sentido ele viu muitos exemplos de programas e de textos associados a esses programas, e ele entende um pouco de misturar soluções específicas que ele já viu com a instrução que vc passou para ele.

Mas não há garantias que o código esta 100% certo.

- vc precisa testá-lo
- provavelmente o ChatGPT viu mais programas pequenos do que grandes, e deve acertar mais em problemas/programas pequenos do que grandes
- para problemas grandes você vai querer dividir/decompor o problema em partes menores nas quais o ChatGPT provavelmente devolverá um programa mais certo, e combinar essas partes para obter a solução final.

Qual a taxa de erro do ChatGPT? qual desses IA generativos é mais “inteligente” para programação? Não sei.

Finalmente, descreveremos muitos dos comandos do Python para que vc seja capaz de entender (parcialmente talvez) a solução que o ChatGPT te devolve.

Avaliação

A avaliação será realizada através de tarefas práticas, ou seja, programas em Python que deverão ser implementados e submetidos pelo Classroom.

Serão propostas n tarefas práticas, que deverão ser realizadas pelos alunos. Os programas desenvolvidos serão testados com um conjunto pré-determinado de testes, que é **desconhecido** dos alunos. A nota de cada atividade prática será proporcional ao número de testes que executarem corretamente.

As tarefas práticas deverão ser resolvidas **durante** as aulas de laboratório (talvez com uma ou duas exceções).

Juntamente com o enunciado de cada tarefa prática será indicado o peso $P_i \in \{1, 2, 3, 4, 5\}$ 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 será aprovado com média final (F) igual a 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:

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

Ou seja, mesmo que o aluno tire uma excelente nota no exame, ele será aprovado mas com a média final 5. Se a nota no exame não for suficiente, o aluno reprovará com a média de E e P.

Caso $P < 2.5$

Neste caso o aluno não pode fazer o exame, e reprovará com a média final igual a P

Observações

- Não haverá atividades práticas substitutivas.
- 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 programa que não siga as condições explicitamente indicadas no enunciado da tarefa (por exemplo, submissão de uma função recursiva válida que resolva o problema).
- Casos de tentativa de plágio ou fraude nas tarefas práticas poderão ser detectadas automaticamente entre todas as submissões das turmas.
- Todos os casos de tentativa de plágio ou fraude automaticamente detectados serão verificados manualmente pela coordenação da disciplina até o final do semestre.

Datas das tarefas práticas

- 18/1
- 25/3
- 8/4
- 15/4
- 22/4
- 29/4

A data das outras atividades práticas ainda serão definidas e os alunos serão avisados em tempo.

Link importante

Esta turma de MC102 não seguirá o plano de aulas, e as atividades práticas das outras turmas de MC102. Assim você não terá muito apoio de outros alunos de outras graduações que também estão fazendo MC102. Mas muitas das aulas da nossa turma são baseadas nas aulas de MC102 para as outras.

Além disso, o site de MC102 tem uma coleção de links para cursos e vídeos de aulas de Python, em português e em inglês.

O link para o site de MC102 é <https://ic.unicamp.br/~mc102/>