

# BCM0505 - Processamento de Informação (Teoria e Prática)

## Plano de ensino

Docentes: Diogo S. Martins, Carla N. Lintzmayer e Paulo H. Pisani  
Centro de Matemática, Computação e Cognição  
Universidade Federal do ABC

Q1 2019  
v. 11/02

### 1 Informações básicas

- TPI: 3-2-5
- Aulas teóricas:
  - Turmas A1D, A2D, A3D:

terças	10-12h	A101-0	quinzenal I
quintas	08-10h	A101-0	semanal
- Aulas Práticas:
  - Turma A1D (Prof. Diogo):

segundas	10-12h	L501	semanal
----------	--------	------	---------
  - Turma A2D (Profa. Carla):

segundas	10-12h	404-2	semanal
----------	--------	-------	---------
  - Turma A3D (Prof. Paulo):

segundas	10-12h	L503	semanal
----------	--------	------	---------
- Plantão de dúvidas:
  - Prof. Diogo:

terças	8-10h	528-2	semanal
--------	-------	-------	---------
  - Profa. Carla:

sextas	10-12h	508-2	semanal
--------	--------	-------	---------
  - Prof. Paulo:

quartas	15-16h	507-2	semanal
---------	--------	-------	---------
- Worskite Tidia: PI-2019Q1-A123D  
Acesso aos materiais de aula
- Atendimento:
  - Sala Piazza.  
Link de *inscrição*: [piazza.com/ufabc.edu.br/spring2019/pi2019q1a123d](https://piazza.com/ufabc.edu.br/spring2019/pi2019q1a123d)  
Link de *acesso*: [piazza.com/ufabc.edu.br/spring2019/pi2019q1a123d/home](https://piazza.com/ufabc.edu.br/spring2019/pi2019q1a123d/home)  
Usaremos Piazza para comunicação entre os participantes (docentes e discentes) da disciplina. Para esclarecer dúvidas sobre o conteúdo, recomenda-se postar as perguntas no Piazza: desse modo, as discussões são socializadas e todos podem se beneficiar.

– Por email (caso não seja possível usar Piazza):

Prof. Diogo: [santana.martins@ufabc.edu.br](mailto:santana.martins@ufabc.edu.br)

Profa. Carla: [carla.negri@ufabc.edu.br](mailto:carla.negri@ufabc.edu.br)

Prof. Paulo: [paulo.pisani@ufabc.edu.br](mailto:paulo.pisani@ufabc.edu.br)

– Presencialmente: no plantão de dúvidas, ou com aviso prévio via email se for fora do plantão.

• Repositório Bitbucket: [https://bitbucket.org/diogo\\_martins/pi-2019q1-a123d/src](https://bitbucket.org/diogo_martins/pi-2019q1-a123d/src)

• URI online judge: <https://www.urionlinejudge.com.br>

Para informações de acesso, vide o email de convite enviado à sua conta institucional ou consulte as instruções fornecidas pelo respectivo professor de prática

Acesso aos códigos-fonte de apoio às aulas.

## 2 Descrição da disciplina

A habilidade de programar computadores é essencial para todo estudante de ciência e tecnologia pois, ao lado da teoria e da experimentação, a computação tornou-se imprescindível para os avanços científicos e tecnológicos. O objetivo da disciplina de Processamento da Informação é preparar os alunos com conceitos e técnicas para processar informação digital eficientemente por meio de programas de computador. A disciplina é introdutória à programação de computadores, com foco no paradigma de programação estruturada/procedural, tomando como base a linguagem de programação Python. A parte teórica tende a focar nos aspectos *conceituais* da programação de computadores, por exemplo: abstração e modelagem de problemas, construção de algoritmos, execução de testes de algoritmos, etc. A parte prática tende a enfatizar os aspectos *lógicos e físicos* da programação de computadores, por exemplo: uso de compiladores, testes online com conjuntos de dados sintéticos ou reais, ferramentas e técnicas de depuração, etc.

## 3 Requisitos recomendados

• BC0005 - Bases Computacionais da Ciência

## 4 Objetivos

- Conhecer os fundamentos de manipulação e tratamento da informação;
- Familiarizar-se com os conceitos de lógica de programação de computadores;
- Conhecer o paradigmas estruturado e procedural de programação;
- Conhecer estruturas de dados fundamentais;
- Saber transferir os conceitos estudados para práticas de programação.

Ao final do curso espera-se que o aluno, aprovado com conceito satisfatório, esteja familiarizado com os conceitos fundamentais de programação estruturada e procedural, sendo capaz de projetar algoritmos para problemas básicos de processamento de informação.

## 5 Ementa

1. Noções de organização de computadores;
2. Lógica de programação, algoritmos e programação (teoria e prática);
3. Sequenciamento de operações;
4. Estruturas de decisão;

5. Estruturas de repetição;
6. Modularização e abstração de dados;
7. Processamento de vetores e matrizes.

## 6 Bibliografia

Obras principais:

1. **Sedgewick, Robert; Wayne, Kevin. Introduction to Programming in Python: An Interdisciplinary Approach. (1st Ed.). Addison-Wesley Professional. 2015. 784 pages.**

Será a nossa obra principal. Na biblioteca, temos apenas a versão em Java, porém os conceitos abordados são praticamente os mesmos. Adicionalmente, você pode acessar o livro na biblioteca online Safari<sup>1</sup> por um período trial de 10 dias.

2. Mark Lutz. Learning Python. (5th Ed.). O'Reilly Media. 2013. 1648 pages.

Apresenta um foco mais “profissional” e menos “acadêmico”, portanto usaremos para questões técnicas mais pontuais. Um ponto positivo é que temos múltiplas cópias na biblioteca.

3. John V. Gutag. Introduction to Computation and Programming Using Python (2nd. Ed.). MIT Press. 2016. 466 pages.

Um pouco mais avançado que os outros, usaremos como bibliografia complementar, caso necessário aprofundar em algum tema.

Vide a última versão do projeto pedagógico do BC&T<sup>2</sup> para as obras complementares dessa disciplina.

## 7 Critérios de avaliação

A avaliação consiste nos componentes dados pela Equação 1, onde:

$$N_F = 0.6 \cdot N_T + 0.4 \cdot N_P \quad (1)$$

- $N_F$  é a nota final;
- $N_T$  é a nota de teoria;
- $N_P$  é a nota de prática.

O conceito final será obtido de acordo com a Equação 2.

$$C_F = \begin{cases} \text{A, se } N_F \in [8.5, 10.0] \\ \text{B, se } N_F \in [7.0, 8.5) \\ \text{C, se } N_F \in [5.5, 7.0) \\ \text{D, se } N_F \in [5.0, 5.5) \\ \text{E, se } N_F \in [0.0, 5.0) \\ \text{O, se ausência total exceder 25\%} \end{cases} \quad (2)$$

**Importante:** O responsável pela digitação das notas de PI é o professor de teoria o qual, ao final do quadrimestre, combina as notas de teoria e de prática para a obtenção do conceito final.

<sup>1</sup><https://www.safaribooksonline.com/library/view/introduction-to-programming/9780134076539/>

<sup>2</sup>Projeto Pedagógico do BC&T UFABC: <http://prograd.ufabc.edu.br/bct/pps>

## 7.1 Avaliação de teoria

A nota de teoria é determinada de acordo com a Equação 3.

$$N_T = 0.4 \cdot N_{p1} + 0.6 \cdot N_{p2} \quad (3)$$

- $N_{p1}$  é a nota da prova 1;
- $N_{p2}$  é a nota da prova 2.

Não teremos atividades avaliativas para entregar (e.g. listas, trabalhos, etc.). Serão publicadas listas de exercícios para subsidiar o estudo, porém essas listas não precisarão ser entregues.

## 7.2 Avaliação de prática

A avaliação consiste nos componentes dados pela Equação 4, onde:

$$N_P = 0.4 \cdot N_{p1} + 0.6 \cdot N_{p2} \quad (4)$$

- $N_{p1}$  é a nota da prova 1;
- $N_{p2}$  é a nota da prova 2.

Teremos práticas de programação, em laboratório ou para estudo individual, com correção automática via *online judge*. As atividades não valem nota, visto que o objetivo principal é prover ao aluno uma lista de problemas representativos dos conceitos vistos na semana, associado a um recurso para verificar automaticamente a validade de suas soluções.

## 7.3 Critérios de avaliação para programas

Os programas solicitados em atividades de avaliação, tanto de teoria quanto de prática, serão analisados quanto aos seguintes critérios:

- **Eficiência:** os programas desenvolvidos deverão ter bom desempenho, o que pode englobar o tratamento adequado dos seguintes fatores:
  - ler e escrever dados nas quantidades mínimas necessárias para resolver o problema;
  - não desperdiçar memória primária (RAM);
  - entre outros.
- **Acurácia:** o programa deverá atender adequadamente a todos os requisitos enunciados para a atividade;
- **Estrutura e organização do código:** atentar principalmente aos seguintes aspectos:
  - **Auto-documentação:** nomes intuitivos para variáveis e métodos/funções;
  - **Modularização:** funções/métodos com alta concisão e baixo acoplamento, isto é, que sejam em sua maioria curtos, e que realizem preferencialmente uma única tarefa;
  - **Comentários:** documentação completa porém ao mesmo tempo concisa (sem poluição visual, apenas nos lugares adequados e necessários). Programar é redigir numa linguagem formal de alto nível, que será interpretada tanto por computadores quanto por humanos.

## 7.4 Mecanismos de avaliação substitutivos

A prova substitutiva será aplicada ao aluno que atender às seguintes condições simultaneamente: *i)* possuir pelo menos 75% de participação; e *ii)* possuir justificativa de ausência em uma das provas. A listagem dos documentos aceitos como justificativa consta na resolução ConsEPE nº 227<sup>3</sup>.

A nota obtida na prova substitutiva necessariamente substituirá a prova para a qual o aluno tem justificativa. A prova substitutiva de teoria será dia 25/04/2019. A data da prova substitutiva de prática deve ser negociada juntamente com o professor da turma de prática, pois não está prevista no calendário da disciplina.

<sup>3</sup><http://prograd.ufabc.edu.br/normas>

## 7.5 Mecanismo de recuperação

A recuperação será aplicada apenas aos alunos que tiverem  $N_F < 5.5$  (i.e.  $C_F = D$  ou  $C_F = F$ ). Consistirá numa prova, em formato similar às aplicadas ao longo do curso. O conteúdo da prova englobará todos os temas vistos durante o quadrimestre.

A nota obtida na prova de recuperação ( $N_R$ ) será usada obter a nota final com recuperação ( $N_{FR}$ ), que consiste na média estabelecida pela Equação 5.

$$N_{FR} = \frac{N_F + N_R}{2} \quad (5)$$

O conceito final com recuperação ( $C_{FR}$ ) é determinado de acordo com os limiares para a nota final de recuperação ( $N_{FR}$ ) dados pela Equação 6.

$$C_{FR} = \begin{cases} C, & \text{se } N_{FR} \geq 5.5 \\ D, & \text{se } N_{FR} \in (5.0, 5.5) \\ F, & \text{se } N_{FR} \leq 5.0 \end{cases} \quad (6)$$

## 8 Cronograma de aulas

O plano a seguir pode variar de acordo com o aproveitamento aferido nas turmas durante o quadrimestre e/ou situações imprevistas. Consulte sempre a versão online, no Tidia, para ter acesso à versão mais recente. A semana 13 corresponde à semana de reposição de feriados.

Sem.	Prática		Teoria	
	Dia	Tema	Dia	Tema
1	11/02	Introdução ao ambiente de desenvolvimento	12/02	Introdução à programação de computadores
			14/02	Tipos de dados, variáveis e programas sequenciais
2	18/02	Exercícios: programas sequenciais	21/02	Estruturas de seleção I
3	25/02	Exercícios: estruturas de seleção	26/02	Estruturas de seleção II
4	04/03	Feriado	28/02	Estruturas de repetição I
			07/03	Estruturas de repetição II
5	11/03	Exercícios: estruturas de repetição	12/03	Revisão
6	18/03	Prova 1	14/03	Prova 1
			23/03	Subprogramas I
7	25/03	Exercícios: subprogramas	26/03	Subprogramas II
8	01/04	Exercícios: vetores	28/03	Vetores
			04/04	Matrizes
9	08/04	Feriado	06/04	Reposição <sup>4</sup> (ref. 09/04 e 11/04): Exercícios de subprogramas, vetores e matrizes
			09/04	Sem aula: afastamento
10	15/04	Exercícios: matrizes	11/04	Sem aula: afastamento
			18/04	Revisão
11	22/04	Exercícios de revisão	23/04	Prova 2
12	29/04 07/05	Prova 2 Vista de provas	25/04	Prova substitutiva
			02/05	Vista de provas + plantão de dúvidas
13			10/05	Prova de recuperação

<sup>4</sup>Sábado, 8-12h, mesma sala de teoria.

## 9 Código de honra

A aprovação na disciplina é baseada exclusivamente no esforço e trabalho pessoal do discente, ao qual cabe garantir que não ajudará ou receberá ajuda não-permitida em qualquer atividade usada pela equipe docente para fins de avaliação (e.g. provas, trabalhos, listas, etc.).

Exemplos de violação do código de honra incluem:

- Copiar atividades avaliativas (e.g. listas, trabalhos, provas, etc.) ou permitir que outros discentes copiem suas atividades avaliativas;
- Colaboração não-permitida entre indivíduos ou grupos (e.g. oferecer vantagens em troca de soluções prontas, doar trechos para o trabalho de outro grupo, etc.);
- Permitir que outros assumam sua identidade em atividades avaliativas (e.g. entregar trabalho que não fez ou permitir que outros façam provas por você);
- Plágio (i.e. aplicável a textos, programas de computador, etc.);
- Receber ou conceder ajuda em atividades avaliativas quando o contexto mostra que não é sensato receber tal ajuda.

Como consequências de violação do código de honra tem-se:

- Reprovação automática na disciplina, com conceito F;
- Denúncia na Comissão de Transgressões Disciplinares Discentes da Graduação, a qual decidirá sobre a punição adequada à violação, o que pode levar a advertência, suspensão ou desligamento, de acordo com os arts. 78-82 do Regimento Geral da UFABC.