

# BCM0505-15 - Processamento da Informação

Feb 4, 2020

## Estudo Continuoado Emergencial (ECE)

A resolução oficial do ECE pode ser acessada [aqui](#). O ECE de PI contempla:

- 7 semanas (de 20/04 a 06/06) de atividades à distância – aulas, laboratórios, listas de exercícios – e atendimentos *online*;
- 3 semanas (início a ser marcado) presenciais para realização de provas, substitutivas e recuperações.

### Aulas Teóricas: Turmas NA{1,2,3} e NB{4,5,6}

Em 5 semanas de curso, tivemos 6 aulas e uma lista de exercícios ( $L_1$ ).

A  $L_1$  não será descartada e fará parte da sua avaliação; veja seção *Critérios de Avaliação* abaixo.

Durante o ECE, teremos 11 aulas (como planejadas abaixo), 2 listas de exercícios ( $L_2$  e  $L_3$ ), uma prova ( $P_1$ ), uma substitutiva ( $P_S$ ), e uma recuperação ( $P_R$ ). As três provas serão presenciais e ocorrerão após o término da quarentena.

O conteúdo de cada aula será disponibilizado e trabalhado via:

- Notas de aula (continuarei escrevendo-as; acessíveis no *link* na identificação de cada aula);
- Coleção de vídeos pequenos (máximo de 20min cada) disponíveis no canal [Aritanan Gruber](#) para PI-ECE 2020.1 no [YouTube](#). Cada vídeo capturará uma tela de computador em que discutiremos um problema, uma ideia de solução e códigos em Python. Vídeos com explicações extras em lousa podem vir a ser usados caso necessário. Vou tentar equilibrar a qualidade versus o tamanho (em *megabytes*) de cada vídeo.
- Encontros *online* via [Google Meet](#) para soluções de dúvidas e esclarecimentos. Tais encontros ocorrerão às quartas e sextas das 19h às 21h. Um *link* para acessar a “sala de discussões” será disponibilizado abaixo com 10min de antecedência.
- Dúvidas também poderão ser enviadas via e-mail para [aritanan.gruber@ufabc.edu.br](mailto:aritanan.gruber@ufabc.edu.br). O tempo de resposta dependerá do número de solicitações e da natureza do conteúdo.

Aula	Data	Conteúdo
<a href="#">A01</a>	22/4 (Qua)	<b>Foco:</b> Revisão I. - Tipos de dados, variáveis e operadores. - Estruturas de seleção: <code>if</code> , <code>elif</code> , <code>else</code> . - Estruturas de repetição: <code>for</code> , <code>while</code> , <code>break</code> , <code>continue</code> , <code>else</code> . - Funções ( <code>def</code> , <code>return</code> ). - Conjuntos, listas e tuplas.
<a href="#">A02</a>	24/4 (Sex)	<b>Foco:</b> Revisão II. - Listas e <i>arrays</i> (vetores): definição, acesso e varredura ( <code>[]</code> , <code>[i:j:k]</code> , <code>len</code> , <code>zip</code> , <code>enumerate</code> , <code>reverse</code> ). - Listas definidas por predicados lógicos ( <i>generators</i> e <i>list comprehensions</i> ). - Concatenação, inserção e remoção: <code>+</code> , <code>append</code> , <code>insert</code> , <code>pop</code> , <code>del</code> . - Operações: <code>in</code> , <code>all</code> , <code>any</code> , <code>min</code> , <code>max</code> , <code>sum</code> , <code>count</code> , <code>index</code> , <code>sort</code> , <code>sorted</code> , <code>map</code> , <code>filter</code> .
<a href="#">A03</a>	01/5 (Sex)	<b>Foco:</b> Permutações em vetores. - Permutações cíclicas, alternantes e adjacentes. - Enumeração de permutações em ordem lexicográfica. - Geração de uma permutação aleatória de forma uniforme.
<a href="#">A04</a>	06/5 (Qua)	<b>Foco:</b> Estatísticas em vetores. - Momentos: média, variância e desvio padrão. - Regressão linear em uma variável. - <a href="#">Heaps</a> e filas de prioridade.
<a href="#">A05</a>	08/5 (Sex)	<b>Foco:</b> Polinômios em vetores. - Polinômios em uma variável: avaliação, adição, multiplicação, derivação e integração simbólicas. - Aplicações e método de <a href="#">Newton-Raphson</a> para cálculo de raízes.
<a href="#">A06</a>	15/5 (Sex)	<b>Foco:</b> Matrizes. - Definição, soma, multiplicação por escalar, transposição e teste de simetria. - Produtos simples, de Harnard e de Krönecker. - Potenciação e exponenciação. - Matrizes em <a href="#">numpy</a> .
<a href="#">A07</a>	20/5 (Qua)	<b>Foco:</b> Matrizes. - Sistemas lineares e <a href="#">eliminação gaussiana</a> . - Cálculo de <a href="#">auto-valores e auto-vetores</a> . - Google's <a href="#">page rank</a> (breve descrição).
<a href="#">A08</a>	22/5 (Sex)	<b>Foco:</b> Matrizes. - <a href="#">Interpolação</a> de pontos no plano por polinômios em uma variável. - <a href="#">Clusterização</a> aproximada de pontos.
<a href="#">A09</a>	29/5 (Sex)	<b>Foco:</b> Grafos em matrizes. - Matriz de adjacências. - Busca em largura e caminhos mínimos. - Algoritmos de <a href="#">Dijkstra</a> e <a href="#">Floyd-Warshall</a> .

Aula	Data	Conteúdo
<a href="#">A10</a>	03/6 (Qua)	<b>Foco:</b> Aprendizado em <a href="#">redes profundas</a> ( <i>deep networks</i> ) I. - Teoria simplificada. - Algoritmo de <a href="#">propagação direta</a> . - <a href="#">Gradiente descendente estocástico</a> .
<a href="#">A11</a>	05/6 (Sex)	<b>Foco:</b> Aprendizado em <a href="#">redes profundas</a> ( <i>deep networks</i> ) II. - Algoritmo de <a href="#">propagação reversa</a> . - Aplicações.

As listas  $L_2$  e  $L_3$  deverão ser entregues via [Tidia](#) até às 19h das datas limites (especificadas abaixo), em um **único** arquivo *pdf*. Atrasos de até 2 dias sofrerão um decréscimo de 20% na nota. Não serão aceitas entregas com atraso maior.

- $L_2$  -> 16/05 (Solução)
- $L_3$  -> 06/06 (Solução)

Seja  $X$  a maior data entre 05/6 e o término oficial da quarentena na UFABC. As três provas serão realizadas em sala, de forma individual e sem consulta, de acordo com o calendário:

- $P_1$  -> primeira sexta-feira após  $X$ , no horário da sua aula antes do ECE
- $P_S$  -> segunda sexta-feira após  $X$ , no horário da sua aula antes do ECE
- $P_R$  -> terceira sexta-feira após  $X$ , no horário da sua aula antes do ECE

### Laboratórios: Turmas NA1 e NB{4,5}

Em 5 semanas de curso, nas turmas NA1 e NB4, tivemos 4 laboratórios e 4 avaliações de laboratório ( $E_1, E_2, E_3$  e  $F_1$ ; esta última será renomeada para  $E_4$ ). As atividades **não** serão descartadas e farão parte da sua avaliação; veja seção *Critérios de Avaliação* abaixo.

A profa. [Cristiane M. Sato](#) não teve condições de aderir ao ECE. Logo, a turma NB5 também ficará sob minha responsabilidade durante o ECE. Esta turma fez 24 exercícios no laboratório / em casa. Estes **não** serão descartados e farão parte da sua avaliação; veja seção *Critérios de Avaliação* abaixo.

Durante o ECE, teremos 7 laboratórios (como planejadas abaixo), 7 avaliações à distância ( $E_5, E_6, \dots, E_{11}$ ; uma por semana), e uma prova presencial ( $P_L$ ) – a ocorrer após o término da quarentena.

- Os exercícios, conjuntamente com cada avaliação, serão disponibilizado no *link* correspondente a cada laboratório;
- Coleção de vídeos pequenos (máximo de 20min cada) disponíveis no canal [Aritanan Gruber](#) para PI-ECE 2020.1 no [YouTube](#).
- Encontros *online* via [Google Meet](#) para soluções de dúvidas e esclarecimentos.  
Tais encontros ocorrerão às terças das 19h às 21h.  
Um *link* para acessar a “sala de discussões” será disponibilizado abaixo com 10min de antecedência.  
Poderemos utilizar o [repl.it](#) para escrevermos código em tempo real.
- Dúvidas também poderão ser enviadas via e-mail para [aritanan.gruber@ufabc.edu.br](mailto:aritanan.gruber@ufabc.edu.br)  
O tempo de resposta dependerá do número de solicitações e da natureza do conteúdo.

Lab	Data	Conteúdo
<a href="#">L01</a>	21/4 (Ter)	Exercícios sobre laços. - < a serem determinados > <b>Avaliação <math>E_5</math></b>
<a href="#">L02</a>	28/4 (Ter)	Exercícios sobre listas. - < a serem determinados > <b>Avaliação <math>E_6</math></b>
<a href="#">L03</a>	05/5 (Ter)	Exercícios sobre listas. - < a serem determinados > <b>Avaliação <math>E_7</math></b>
<a href="#">L04</a>	12/5 (Ter)	Exercícios sobre listas e matrizes. - < a serem determinados > <b>Avaliação <math>E_8</math></b>
<a href="#">L05</a>	19/5 (Ter)	Exercícios sobre matrizes. - < a serem determinados > <b>Avaliação <math>E_9</math></b>
<a href="#">L06</a>	26/5 (Ter)	Exercícios sobre matrizes. - < a serem determinados > <b>Avaliação <math>E_{10}</math></b>
<a href="#">L07</a>	02/5 (Ter)	Exercícios sobre matrizes. - < a serem determinados > <b>Avaliação <math>E_{11}</math></b>

Das sete *mini-avaliações* ( $\{E_j\}_{j \in \{5,6, \dots, 11\}}$ ), 4 serão selecionadas pseudo-aleatoriamente (seleção única, não individual) e corrigidas. As notas resultantes ( $E_j \in [0, 2.5]$ ) serão utilizadas no cômputo da sua avaliação de laboratório. Como você não saberá à priori quais 4 das 7 mini-avaliações serão escolhidas, dedique-se à todas com diligência e profissionalismo.

Para as turmas NA1 e NB4: uma nota  $F = (E_3 + E_4)/2 \in [0, 10]$  será atribuída a você como parte da sua avaliação pré-ECE.

Para a turma NB5: a prof. Cris fornecerá a nota  $F$  baseada nos exercícios que você fez com ela.

Seja  $X$  a maior data entre 05/6 e o término oficial da quarentena na UFABC. A prova de laboratório será realizada, de forma individual e sem consulta, de acordo com o calendário:

- $P_L$  -> primeira terça-feira após  $X$ , no horário do seu lab antes do ECE

Caso algum aluno falte na  $P_L$ , ocorrerá uma substitutiva (fechada) a ela na segunda terça-feira após  $X$  no mesmo horário.

## Critérios de Avaliação

Sua nota será composta por:

Avaliação	Peso	Filtragem
Listas de exercícios ( $L$ )	20% (10% cada)	[ duas melhores dentre três ]
Provas ( $P$ )	40%	[ melhor dentre duas ]
Laboratório ( $E$ )	40% total	[ de acordo com seu professor ]

Em detalhes:

- $L = \max\{L_1 + L_2, L_1 + L_3, L_2 + L_3\}/2$
- $P = \max\{P_1, P_S\}$
- $E = \text{como definido para a sua turma}$   
para NA1, NB4 e NB5

$$E = \frac{1}{3}P_L + \frac{1}{3}F + \frac{1}{3}\sum_{j \in S} E_j$$

em que  $S \subseteq \{5, 6, \dots, 11\}$  e  $|S| = 4$ , e  $F$  é a sua nota de laboratório pré-ECE.

**Nota nominal:**  $N = 0.2 * L + 0.4 * P + 0.4 * E$

**Conceito nominal:**  $C_N$  é obtido por  $N \in [a . . b]$  como na tabela abaixo:

Conceito $C_N$	Intervalo
A	(8.5, +∞]
B	(7.0, 8.5]
C	(6.0, 7.0]
D	[5.0, 6.0]
F	[−∞, 5.0)

O conceito nominal reflete o seu desempenho frente ao material apresentado e às avaliações realizadas.

**Normalização:** sejam  $\mu$  e  $\sigma$  a média e o desvio padrão, respectivamente, das notas  $N$  atribuídas a todos os alunos das 6 turmas. Cada aluno obterá uma **nota normalizada**:

$$M = (N - \mu)/\sigma$$

Um **conceito normalizado**  $C_M$  será atribuído de acordo com o encaixe de  $M \in [a . . b]$  na tabela abaixo:

Conceito $C_M$	Intervalo
A	(3σ/2, +∞]
B	(σ, 3σ/2]
C	(σ/2, σ]
D	[0, σ/2]
F	[−∞, 0)

O conceito normalizado reflete o seu desempenho perante os seus colegas.

Considerando-se a ordenação  $A > B > C > D > F$ , seu conceito (final / pré-recuperação) será:

$$C = \max\{C_N, C_M\}$$

Ou seja, seu conceito (final / pré-recuperação) será maior ou igual ao seu conceito nominal.

**Recuperação:**

Caso seu conceito  $C$  seja D ou F, você tem direito a uma prova de recuperação  $P_R$ . Esta será única e contemplará toda a matéria do quadrimestre. Uma nova nota nominal  $\bar{N} = (N + P_R)/2$  será utilizada para gerar novos:

- conceito nominal  $\bar{C}_N$ ,
- nota normalizada  $\bar{M} = (\bar{N} - \mu)/\sigma$ , e
- conceito normalizado  $\bar{C}_M$

de acordo com as regras apresentadas. Seu conceito final será então  $\bar{C} = \max\{\bar{C}_N, \bar{C}_M\}$ . Observe que:

- serão utilizados os parâmetros  $\mu$  e  $\sigma$  computados pré-recuperação (o que não te penaliza em caso de melhora);
- seu conceito final pós-recuperação pode ser menor que o pré-recuperação: uma vez feita, a recuperação é parte integrante da sua avaliação.

## Integridade acadêmica e transgressões

O Artigo 25 do [Código de Ética da UFABC](#) estabelece, à página 23: “Quanto aos trabalhos acadêmicos, é eticamente inaceitável que os discentes:

- I - fraudem avaliações;
- II - fabriquem ou falsifiquem dados;
- III - plajeiem ou não creditem devidamente autoria;
- IV - aceitem autoria de material acadêmico sem participação na produção;
- V - vendam ou cedam autoria de material acadêmico próprio a pessoas que não participaram da produção.”

Trabalhos (listas, provas, programas) suspeitos de cópia ou de outra representação fraudulenta acarretarão aos envolvidos conceitos F (falha) no curso. A atividade será reportada à [Comissão Disciplinar Discente](#) da universidade para que sejam tomadas todas as providências disciplinares cabíveis.

## Notas e conceitos

Lista em ordem crescente dos últimos 5 dígitos do seu RA seguido pela primeira letra do seu último sobrenome.

- Versão de [20/04/2020](#).

---

## O restante desta página contém o conteúdo original (pré-ECE) para consulta

### Descrição e objetivos

Introdução ao *Pensamento Computacional* e à arte de resolver *Problemas Algorítmicos* em dispositivos [clássicos](#) via programação em [Python](#). Ênfase na obtenção de soluções construtivas para problemas elementares de natureza discreta ou contínua.

O curso é essencialmente auto-contido, mas é recomendado que as disciplinas [Bases Matemáticas](#) e [Natureza da Informação](#) tenham sido cursadas à priori. Ao final, a/o estudante aprovada/o deve ser capaz de desenvolver soluções algorítmicas simples, baseando-se nos rudimentos de análise e programação apresentados.

Alguns dos tópicos contemplados: variáveis, constantes e palavras reservadas; sintaxe e semântica; tipos de dados fundamentais e operadores; estruturas condicionais simples e compostas; funções e módulos; estruturas de repetição (iteração e recursão); listas, filas e pilhas; algoritmos elementares de busca, separação e ordenação; cálculo de estatísticas simples; vetores, matrizes e tensores; computação numérica para zeros de polinômios, interpolação, regressão, integração e sistemas lineares; aplicações em clustering, neural networks, page ranking e caminhos mínimos.

### Docentes e links

E-mail	Sítios	Sala	Atendimento
<a href="#">Aritanan Gruber</a>	( <a href="#">aud+lab</a> ; <a href="#">Tidia</a> )	S.539-2, bloco A (SA)	Terças e Sextas 16h30–18h30h
<a href="#">Cristiane M. Sato</a>	( <a href="#">lab</a> )	S.539-2, bloco A (SA)	Terças 17h30–18h30
<a href="#">Diogo S. Martins</a>	( <a href="#">Tidia</a> )	S.582-2, bloco A (SA)	Terças 18–19h
<a href="#">Luiz C. Rozante</a>	( <a href="#">Tidia</a> )	S.548-2, bloco A (SA)	Terças 18–19h
<a href="#">Valério R. Batista</a>	( <a href="#">Tidia</a> )	S.818-B, bloco B (SA)	Terças 16–17h

### Aulas

#### Teóricas: algoritmos (auditório)

Turmas	Quartas I	Sextas I e II	Professor
NA1, NA2, NA3	21–23h (A.101-0)	19–21h (A.101-0)	Aritanan Gruber
NB4, NB5, NB6	19–21h (S.208-0)	21–23h (S.213-0)	Aritanan Gruber

#### Práticas: programação (laboratório)

Turma	Terças I e II	Professor
NA1	21–23h (L.501-B)	Aritanan Gruber
NA2	21–23h (S.404-2)	Luiz C. Rozante
NA3	21–23h (L.502-B)	Valério R. Batista
NB4	19–21h (S.402-2)	Aritanan Gruber
NB5	19–21h (S.403-2)	Cristiane M. Sato
NB6	19–21h (L.703-B)	Diogo S. Martins

A presença às aulas é mandatória e a/o aluna/o é responsável pelo aprendizado de toda informação apresentada em cada aula.

## Bibliografia

A referência básica é Sedgewick-Wayne-Dondero [A], Cormen [B] e Dasgupta-Papadimitriou-Vazirani [C] são boas complementações.

- [A] R. Sedgewick, K. Wayne e R. Dondero, [Introduction to Programming in Python: An Interdisciplinary Approach](#), Pearson/Addison-Wesley Professional (2015)
- [B] T. Cormen, *Algorithms Unlocked*, MIT Press (2013)  
Versão em Português: *Desmistificando Algoritmos: Teoria e Prática*, GEN LTC (2013)
- [C] S. Dasgupta, C. Papadimitriou e U. Vazirani, [Algorithms](#), McGraw-Hill (2006)  
Versão em Português: *Algoritmos*, AMGH (2009)

## Recursos extras

Bentley [D], Kernighan-Pike [E] e Kernighan-Plauger [F] são excelentes recursos sobre programação de um modo geral, focando em aspectos como interpretação, compilação, depuração, ferramentas auxiliares, práticas eloquentes, comparações entre soluções, ideias recorrentes, erros comuns, entre outros.

- [D] J. Bentley, *Programming Pearls*, 2nd Ed., Pearson (1999)
- [E] B. Kernighan e D. Pike, *The Practice of Programming*, Pearson (1999)
- [F] B. Kernighan e P. Plauger, *The Elements of Programming Style*, 2nd Ed., McGraw-Hill (1978)

Harel-Feldman [G] apresenta uma cobertura doce e bem feita do papel exercido pelo conceito algorítmico dentro e fora da ciência da computação, explorando sua vasta aplicabilidade e exibindo suas limitações – leitura altamente recomendada!

- [G] D. Harel e Y. Feldman, *Algorithmics: The Spirit of Computing*, 3rd Ed., Addison-Wesley (2004)

Um enfoque algorítmico mais avançado é encontrado em Cormen-Leiserson-Rivest-Stein [H] e Erickson [I]. Lehman-Leighton-Meyer [K] contém material sobre provas matemáticas, conjuntos, funções e relações, grafos, somatórias, recursões, análise assintótica, e probabilidade discreta. Klein [J] e Mohri-Rostamizadeh-Talwalkar [L] desenvolvem o lado teórico de alguns dos exemplos que serão cobertos.

- [H] T. Cormen, C. Leiserson, R. Rivest, C. Stein, [Introduction to Algorithms](#), 3rd Ed., MIT University Press (2009)  
Versão em Português: *Algoritmos: Teoria e Prática*, Elsevier (2012)
- [I] J. Erickson, [Algorithms](#), gratuito (2019)
- [J] P. Klein, [Coding The Matrix: Linear Algebra Through Computer Science Applications](#), Newtonian Press (2013)
- [K] E. Lehman, F. Leighton e A. Meyer, [Mathematics for Computer Science](#), notas de aula a serem publicadas (2020)
- [L] M. Mohri, A. Rostamizadeh, A. Talwalkar, [Foundations of Machine Learning](#), MIT Press (2012)

Material introdutório e oficial sobre Python é encontrado em [N,O]. O site Geeksforgeeks [M] contém códigos em Python de alguns algoritmos desenvolvidos em sala e em laboratório. Não verifiquei diligentemente todo o conteúdo, logo use-o sob sua discricão.

- [M] Geeksforgeeks, [Fundamentals of Algorithms](#), implementações em C/C++, Java e Python
- [N] Python Software Foundation, [Python Beginner's Guide](#)
- [O] Python Software Foundation, [The Python Tutorial](#)

Relacionados a tópicos cobertos e que valem à pena serem olhados:

- [P] A. Dasdan, [Eleven Simple Algorithms to Compute Fibonacci Numbers](#), arXiv (2018)
- [Q] S. Fekete, S. Morr e S. Stiller, [IDEA: An ongoing series of nonverbal algorithm assembly instructions](#), TU Braunschweig (2018)

## Avaliações

### Listas de exercícios (Teoria)

Entrega via [Tidia](#) até às 09h da data limite, em um **único** arquivo pdf.

Atrasos de até 2 dias sofrerão um decréscimo de 20% na nota. Não serão aceitas entregas com atraso maior.

- $L_1$  -> 10/03/2020 ([Solução](#))
- $L_2$  -> 06/04/2020 (Solução)
- $L_3$  -> 04/05/2020 (Solução)

Colaborações na resolução das listas são válidas e incentivadas, mas as entregas são individuais e cada aluna/o deve escrever suas próprias soluções (não copie!). Nomes de seus colaboradores e fontes consultadas (livros, artigos, páginas web) devem constar no início de cada problema resolvido.

### Provas (Teoria)

Individuais, sem consulta, em auditório.

A substitutiva ( $P_S$ ) será aberta à todos que desejarem fazê-la, sem ônus.

- $P_1$  -> 20/03/2020 (Solução)
- $P_2$  -> 07/05/2020 (Solução)
- $P_S$  -> 08/05/2020 (Solução)
- $P_R$  -> 14/05/2020 (Solução)

### Exercícios programa (Prática: NA1 e NB4)

- Dez *mini-avaliações* ( $\{E_j\}_{j \in [10]}$ ), uma por noite, do segundo ao décimo primeiro laboratório:

- 30–60min para resolução de 1 a 3 problemas e submissão via Tidia (dos programas);
- ao final, 5 serão selecionadas pseudo-aleatoriamente (seleção única, não individual) e corrigidas;
- notas resultantes ( $E_j \in [0, 2]$ ) serão utilizadas no cômputo da sua avaliação de laboratório;
- cada aluna/o tem direito a até 2 substituições (também pseudo-aleatórias) para o caso exclusivo de *faltas*.

Nota: como você não saberá à priori quais 5 das 10 mini-avaliações serão escolhidas, dedique-se à todas com diligência e profissionalismo.

- Cinco avaliações ( $\{F_j\}_{j \in [5]}$ ) para serem feitas *em casa*:
  - submissão via Tidia (dos programas) até a data especificada no mapa das aulas;
  - quatro melhores notas ( $F_j \in [0, 2.5]$ ) serão utilizadas no cômputo da sua avaliação de laboratório.

### Sobre as correções

A menos que explicitado o contrário, seus códigos deverão estar escritos em Python. A correção não é binária (funciona versus não funciona) e leva em consideração a qualidade das suas ideias e o quão perto de uma solução (completa) você chegou. Em avaliações teóricas, pequenos erros sintáticos (que não comprometam a semântica do código) não serão penalizados. Em avaliações práticas, será à critério do seu professor.

Para as turmas NA1 e NB4: seu código deverá ser interpretável (não apresentar erros sintáticos ou em tempo de execução como divisões por 0, acessos a posições indefinidas em listas, etc.) e o resultado correto.

## Critérios

### Sua nota será composta por:

Avaliação	Peso	Filtragem
Listas de exercícios ( $L$ )	20% (10% cada)	[ duas melhores dentre três ]
Provas ( $P$ )	40% (20% cada)	[ duas melhores dentre três ]
Laboratório ( $E$ )	40% total	[ de acordo com seu professor ]

### Em detalhes:

- $L = \max\{L_1 + L_2, L_1 + L_3, L_2 + L_3\}/2$
- $P = \max\{P_1 + P_2, P_1 + P_3, P_2 + P_3\}/2$
- $E = \text{como definido para a sua turma}$   
para NA1 e NB4,

$$E = \frac{3}{4} \sum_{j \in S} E_j + \frac{1}{4} \sum_{j \in T} F_j$$

em que

- $S \subseteq \{1, 2, \dots, 10\}$  e  $|S| = 5$ ,
- $T = \{1, 2, \dots, 5\} \setminus \{i\}$ , com  $i$  sendo o índice de **uma** menor nota em  $\{F_1, F_2, \dots, F_5\}$ .

**Nota nominal:**  $N = 0.2 * L + 0.4 * P + 0.4 * E$

**Conceito nominal:**  $C_N$  é obtido por  $N \in (a, b]$  como na tabela abaixo:

Conceito $C_N$	Intervalo
A	$(8.5, +\infty]$
B	$(7.0, 8.5]$
C	$(6.0, 7.0]$
D	$(5.0, 6.0]$
F	$(-\infty, 5.0]$

O conceito nominal reflete o seu desempenho frente ao material apresentado e às avaliações realizadas.

**Normalização:** sejam  $\mu$  e  $\sigma$  a média e o desvio padrão, respectivamente, das notas  $N$  atribuídas a todos os alunos das 6 turmas. Cada aluno obterá uma **nota normalizada**:

$$M = (N - \mu)/\sigma$$

Um **conceito normalizado**  $C_M$  será atribuído de acordo com o encaixe de  $M \in (a, b]$  na tabela abaixo:

Conceito $C_M$	Intervalo
A	$(3\sigma/2, +\infty]$
B	$(\sigma, 3\sigma/2]$
C	$(\sigma/2, \sigma]$
D	$(0, \sigma/2]$
F	$(-\infty, 0]$

O conceito normalizado reflete o seu desempenho perante os seus colegas.

Considerando-se a ordenação  $A > B > C > D > F$ , seu conceito (final / pré-recuperação) será:

$$C = \max\{C_N, C_M\}$$

Ou seja, seu conceito (final / pré-recuperação) será maior ou igual ao seu conceito nominal.

### Recuperação:

Caso seu conceito  $C$  seja D ou F, você tem direito a uma prova de recuperação  $P_R$ . Esta será única e contemplará toda a matéria do quadrimestre. Uma nova nota nominal  $\bar{N} = (N + P_R)/2$  será utilizada para gerar novos:

- conceito nominal  $\bar{C}_N$ ,
- nota normalizada  $\bar{M} = (\bar{N} - \mu)/\sigma$ , e
- conceito normalizado  $\bar{C}_M$

de acordo com as regras apresentadas. Seu conceito final será então  $\bar{C} = \max\{\bar{C}_N, \bar{C}_M\}$ . Observe que:

- serão utilizados os parâmetros  $\mu$  e  $\sigma$  computados pré-recuperação (o que não te penaliza em caso de melhora);
- seu conceito final pós-recuperação pode ser menor que o pré-recuperação: uma vez feita, a recuperação é parte integrante da sua avaliação.

### Integridade acadêmica e transgressões

O Artigo 25 do [Código de Ética da UFABC](#) estabelece, à página 23: “Quanto aos trabalhos acadêmicos, é eticamente inaceitável que os discentes:

- I - fraudem avaliações;
- II - fabriquem ou falsifiquem dados;
- III - plajeiem ou não creditem devidamente autoria;
- IV - aceitem autoria de material acadêmico sem participação na produção;
- V - vendam ou cedam autoria de material acadêmico próprio a pessoas que não participaram da produção.”

Trabalhos (listas, provas, programas) suspeitos de cópia ou de outra representação fraudulenta acarretarão aos envolvidos conceitos F (falha) no curso. A atividade será reportada à [Comissão Disciplinar Discente](#) da universidade para que sejam tomadas todas as providências disciplinares cabíveis.

### Notas e conceitos

Lista em ordem crescente dos últimos 5 dígitos do seu RA seguido pela primeira letra do seu último sobrenome.

- Versão de [11/02/2020](#).

### Mapa das aulas teóricas em auditório

Para um melhor aproveitamento da disciplina, é recomendada a leitura do material de cada aula antes que ela ocorra.

Aula	Dia	Conteúdo
01	12/2 (Qua)	<p><b>Foco:</b> Leve introdução ao raciocínio algorítmico e um sucinto <i>tour</i> da linguagem Python. (<a href="#">conteúdo</a>)</p> <ul style="list-style-type: none"> <li>- Expediente (listas, provas e critérios).</li> <li>- Um primeiro problema: busca de um valor em uma coleção finita de inteiros. <ul style="list-style-type: none"> <li>- Modelo computacional determinístico sequencial.</li> <li>- Invariantes e um algoritmo iterativo; correção e finitude; número de comparações no pior caso.</li> <li>- Estruturas de programação necessárias para expressarmos o algoritmo (clareza e precisão).</li> <li>- Como Python provê estas estruturas (tipos de dados, variáveis, listas, operadores, seleções, repetições, funções, módulos).</li> <li>- Um código em Python para nosso algoritmo iterativo; simulação de execução para um exemplo.</li> <li>- Interpretação versus compilação; execução de programas no computador.</li> <li>- Variações e uso de sentinelas.</li> </ul> </li> </ul> <p>A partir da próxima aula, vamos investigar mais à fundo cada uma das estruturas de programação providas por Python.</p>
02	14/2 (Sex)	<p><b>Foco:</b> Estruturas de controle de fluxo. (<a href="#">conteúdo</a>)</p> <ul style="list-style-type: none"> <li>- Tipos de dados (<code>int</code>, <code>float</code>, <code>bool</code>, <code>string</code>); constantes e variáveis. <ul style="list-style-type: none"> <li>- Tipagem estática (C/C++) versus tipagem dinâmica (Python).</li> <li>- Conversões de tipo em Python.</li> </ul> </li> <li>- Operadores: <ul style="list-style-type: none"> <li>- Aritméticos (+, -, *, /, //, %, **);</li> <li>- De atribuição (=, +=, -=, *=, /=, //=, %=);</li> <li>- Relacionais (==, !=, &lt;, &lt;=, &gt;, &gt;=);</li> <li>- Lógicos (<code>and</code>, <code>or</code>, <code>not</code>).</li> </ul> </li> <li>- Estruturas de controle de fluxo: seleção (<code>if</code>, <code>elif</code>, <code>else</code>).</li> <li>- Estruturas de controle de fluxo: repetição (<code>while</code>, <code>break</code>, <code>continue</code>).</li> <li>- Funções (<code>def</code>): desenvolvendo código reutilizável.</li> <li>- Um segundo problema: busca em uma coleção finita e <i>ordenada</i> de inteiros. <ul style="list-style-type: none"> <li>- Divisão e conquista; biseção e busca binária; algoritmo iterativo (com código); correção e números de comparações no pior caso (recorrência).</li> </ul> </li> </ul>
03	21/2 (Sex)	<p><b>Foco:</b> Estruturas de controle de fluxo. (<a href="#">conteúdo</a>)</p> <ul style="list-style-type: none"> <li>- Funções (<code>def</code>): passagem de parâmetros por valor e por referência; anotações de tipos.</li> <li>- Módulos (<code>import</code>).</li> <li>- Estruturas de controle de fluxo (<code>if</code>, <code>for</code>, <code>while</code>): <ul style="list-style-type: none"> <li>- Ordenação de 3 elementos.</li> <li>- Cálculo de números Harmônicos e de Fibonacci.</li> <li>- Aproximações de <math>e^x</math> e <math>\cos x</math> via séries de <a href="#">Taylor-Maclaurin</a>.</li> <li>- Teste de primalidade.</li> </ul> </li> </ul>
==	26/2 (Qua)	Recesso de Carnaval. Reposição em 7/5 (Qui).
04	28/2 (Sex)	<p><b>Foco:</b> Estruturas de controle de fluxo. (<a href="#">conteúdo</a>)</p> <ul style="list-style-type: none"> <li>- Fatoração de inteiros.</li> <li>- Crivo de Erathostenes.</li> <li>- Funções como parâmetros.</li> <li>- Derivação numérica: método de diferenças finitas.</li> <li>- Integração numérica: algoritmos de ponto médio, trapézios, e Simpson.</li> </ul>

Aula	Dia	Conteúdo
05	6/3 (Sex)	<b>Foco:</b> Vetores. ( <a href="#">conteúdo</a> ). - Listas e arrays ( <code>[]</code> , <code>len</code> , <code>min</code> , <code>sum</code> , <code>zip</code> , <code>enumerate</code> ). - Produto escalar, norma euclidiana, ângulo entre vetores, distância entre pontos. - Estatísticas elementares: mínimo, segundo mínimo, média, momentos, variância e desvio padrão.
06	11/3 (Qua)	<b>Foco:</b> Vetores. ( <a href="#">conteúdo</a> ). - Listas e arrays ( <code>in</code> , <code>insert</code> , <code>append</code> , <code>pop</code> ). - Variações em buscas sequencial e binária. - Ordenação de listas por seleção, inserção, intercalação e particionamento.
07	13/3 (Sex)	<b>Foco:</b> Vetores. ( <a href="#">conteúdo</a> ). - Listas definidas por predicados lógicos ( <i>list comprehensions</i> ). - Operadores e funções de listas ( <code>all</code> , <code>any</code> , <code>count</code> , <code>filter</code> , <code>index</code> , <code>reverse</code> , <code>sorted</code> ). - Segmentos de listas ( <code>[i:j:k]</code> ). - Exemplos de uso em problemas das aulas passadas.
08	20/3 (Sex)	Prova 1 ( $P_1$ ).
09	25/3 (Qua)	<b>Foco:</b> Vetores. ( <a href="#">conteúdo</a> ). - Polinômios em uma variável: avaliação, adição, multiplicação, derivação e integração simbólicas. - Aplicações e método de para cálculo de raízes.
10	27/3 (Sex)	<b>Foco:</b> <a href="#">Matrizes</a> . - Soma, produto, produtos de Harnard e de Kronecker, teste de simetria, transposição, potenciação, exponenciação.
11	3/4 (Sex)	<b>Foco:</b> Matrizes. - Matriz de adjacências, busca em largura e <a href="#">caminhos mínimos</a> . - Algoritmo de <a href="#">Floyd-Warshall</a> .
==	8/4 (Qua)	Aniversário de Santo André. Reposição em 8/5 (Sex).
==	10/4 (Sex)	Recesso de Páscoa. Reposição em 11/5 (Seg).
12	17/4 (Sex)	<b>Foco:</b> Matrizes. - Sistemas lineares e <a href="#">eliminação gaussiana</a> . - <a href="#">Cálculo</a> de auto-valores e auto-vetores. - Google's <a href="#">page rank</a> (breve descrição).
13	22/4 (Qua)	<b>Foco:</b> Matrizes. - <a href="#">Interpolação</a> de pontos no plano por polinômios em uma variável. - <a href="#">Clusterização</a> aproximada de pontos.
14	24/4 (Sex)	<b>Foco:</b> Recursão. - Problemas auto-redutíveis; indução e algoritmos recursivos; recorrências. - Iteração versus recursão. - Enumeração de <a href="#">permutações</a> . - Soluções recursivas para alguns problemas das aulas anteriores.
==	1/5 (Sex)	Recesso do Dia do Trabalho. Reposição em 14/5 (Sex).
15	7/5 (Qui)	Prova 2 ( $P_2$ ).
16	8/5 (Sex)	Prova Substitutiva ( $P_S$ ).
17	11/5 (Seg)	Aula de revisão / solução de alguns exercícios prévios.
18	14/5 (Sex)	Prova de Recuperação ( $P_R$ ).

## Mapa das aulas práticas em laboratório (NA1 e NB4)

Lab	Dia	Conteúdo
01	11/2 (Ter)	- Expediente (resumido). - <a href="#">Python 3</a> e <a href="#">repl.it</a> . - Entrada e saída ( <code>print</code> e <code>input</code> ). - Variáveis inteiras ( <code>int</code> ) e ponto-flutuante ( <code>float</code> ); operadores. - Estruturas de repetição ( <code>for</code> , <code>range</code> ). - Declarações de função ( <code>def</code> ) e anotações de tipos. - <a href="#">Exercícios</a> .
02	18/2 (Ter)	- Estruturas de seleção ( <code>if</code> , <code>elif</code> , <code>else</code> ); variáveis lógicas ( <code>bool</code> ). - Estruturas de repetição ( <code>while</code> , <code>break</code> , <code>continue</code> ). - <a href="#">Exercícios</a> . - <b>Avaliação <math>E_1</math></b> .
==	25/2 (Ter)	Recesso de Carnaval. Reposição em 6/5 (Qua).
03	3/3 (Ter)	- <a href="#">Exercícios</a> . - <b>Avaliação <math>E_2</math></b> . - <b>Avaliação <math>F_1</math></b> .
04	10/3 (Ter)	- <a href="#">Exercícios</a> . - <b>Avaliação <math>E_3</math></b> .
05	17/3 (Ter)	- Exercícios. regressão linear, par de pontos mais próximo. - <b>Avaliação <math>E_4</math></b> . - <b>Avaliação <math>F_2</math></b> .
06	24/3 (Ter)	- Exercícios. códigos de Huffman, cobertura mínima (aproximada) de conjuntos. - <b>Avaliação <math>E_5</math></b> .



Lab	Dia	Conteúdo
07	31/3 (Ter)	- Exercícios. produto tensorial entre matrizes, quadrados mágicos, estabilidade de rainhas. - <b>Avaliação</b> $E_6$ . - <b>Avaliação</b> $F_3$ .
08	7/4 (Ter)	- Exercícios. percurso em labirintos (especificados em arquivo); função de Grundy e estratégia vencedora em NIN. - <b>Avaliação</b> $E_7$ .
09	14/4 (Ter)	- Exercícios. medidas de centralidade em redes sociais. - <b>Avaliação</b> $E_8$ . - <b>Avaliação</b> $F_4$ .
==	21/4 (Ter)	<i>Recesso de Tiradentes. Reposição em 13/5 (Qua).</i>
10	28/4 (Ter)	- Exercícios. otimização de $\max c^T x : Ax = b, x \geq 0$ . - <b>Avaliação</b> $E_9$ .
11	6/5 (Qua)	- Exercícios. - <b>Avaliação</b> $E_{10}$ . - <b>Avaliação</b> $F_5$ .
12	13/5 (Qua)	Aula de revisão / solução de alguns exercícios prévios para a recuperação ( $P_R$ ).

## Monitorias

- Plantões presenciais de monitores em SA, L.504-B

Horário	Seg	Ter	Qua	Qui	Sex
15–16h	Chung	Chung		Chung	
16–17h	Chung	Chung		Chung	
17–18h	Chung	Pedro		Chung	Pedro
18–19h	Heitor	Pedro		Pedro	Pedro

- Plantões presenciais de monitores em SB, blocos Alfa-1 e Alfa-2

Horário	Seg (A1-L101)	Ter	Qua (A2-L001)	Qui	Sex (A2-L001)
14–15h			André		André
15–16h	André		André		André
16–17h	André				
17–18h	André				
18–19h	André				

- Atendimento online via [facebook](#) – basta acessar o chat a qualquer hora
- Atendimento pró-ativo via [ava](#) para exercícios no *moodle*

## Estudando para esta disciplina

Este curso tem nível introdutório e contém uma coleção de problemas elementares e fundamentais. Apesar disso, é normal fazer confusões e sentir-se *perdido* no início. O motivo é, em geral, a falta de familiaridade com o raciocínio algorítmico – e algo que este curso pretende reverter.

- Refaça os exemplos fornecidos em sala de aula e no laboratório.
- Preste atenção ao processo de solução e não foque somente no resultado final.
- Assista às aulas (faça perguntas), resolva as listas de exercícios e os problemas de laboratório.
- Estude a bibliografia indicada (monte grupos de estudo) e faça um bom uso dos horários de atendimento dos professores.
- Tenha sempre em mente que aprendizado é uma tarefa ativa; não fique somente assistindo.
- Programe, programe, programe. Quanto mais programação você fizer, melhor ficará.
- Se ainda estiver se sentindo *perdido*, repita os passos acima. Mais cedo ou mais tarde, eles convergirão à compreensão.



### Aritanan Gruber

Assistant Professor

“See, if y’all haven’t the same feeling for *this*, I really don’t give a damn. If you ain’t feeling it, then dammit *this* ain’t for you!”  
(desconheço a autoria; agradeço a indicação)

[✉](#) [📄](#) [R<sup>6</sup>](#) [📄](#) [in](#) [📄](#) [cv](#)

