

Caracterização da disciplina							
Código da disciplina:	BCM0505-15	Nome da disciplina:	Processamento da Informação				
Créditos (T-P-I):	(3-2-5)	Carga horária:	60 horas	Aula prática:	N	Câmpus:	SBC
Código da turma:	DB1BCM0505-15SB	Turma:	B2	Turno:	Matutino	Quadrimestre:	2
Docente(s) responsável(is):		RAPHAEL YOKOINGAWA DE CAMARGO (T) / VALERIO BATISTA (P)					

Alocação da turma						
	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
8:00 - 9:00			Teoria			
9:00 - 10:00			Teoria			
10:00 - 11:00				Teoria (sem. I)	Prática	
11:00 - 12:00				Teoria (sem. I)	Prática	
12:00 - 13:00						
13:00 - 14:00						
14:00 - 15:00						
15:00 - 16:00						
16:00 - 17:00						
17:00 - 18:00						
18:00 - 19:00						
19:00 - 20:00						
20:00 - 21:00						
21:00 - 22:00						
22:00 - 23:00						

Planejamento da disciplina			
Objetivos gerais			
Apresentar os fundamentos sobre manipulação e tratamento da Informação, principalmente por meio da explicação e experimentação dos conceitos e do uso prático da lógica de programação.			
Objetivos específicos			
Que o aluno seja capaz de compreender os conceitos fundamentais a respeito da manipulação e tratamento da Informação. Que o aluno entenda a lógica de programação de computadores e adquira a habilidade prática de desenvolver algoritmos básicos para modelar e solucionar problemas de natureza técnico-científica, independentemente de uma linguagem ou de um paradigma de programação específicos.			
Ementa			
Introdução a algoritmos. Variáveis e tipos de dados. Operadores aritméticos, lógicos e precedência. Métodos/Funções e parâmetros. Estruturas de seleção. Estruturas de repetição. Vetores. Matrizes. Entrada e saída de dados. Depuração. Melhores práticas de programação.			
Conteúdo programático			
Aula	Conteúdo	Estratégias didáticas	Avaliação
08/06 (T)	Apresentação da disciplina. Noções básicas de programação de computadores. Noções de algoritmos.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Não há
09/06 (T)	Variáveis. Operadores matemáticos. Operadores de atribuição. Precedência.	Aula expositiva em vídeo. Resolução de	Exercícios de fixação

	Programação de operações em fluxo sequencial.	problemas. Exemplos.	
10/06 (P)	Exercícios de ambientação com programação. Algoritmos sequenciais.	Resolução de problemas. Exemplos.	Exercícios práticos
15/06 (T)	Algoritmos sequenciais com modularização. Módulos. Definição e execução. Parametrização.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
17/06 (P)	Recesso	Recesso	Recesso
22/06 (T)	Estruturas de controle de fluxo. Estruturas condicionais. Operadores comparativos e lógicos.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
23/06 (T)	Estruturas de repetição. Algoritmos envolvendo repetição.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
24/06 (P)	Exercícios com modularização e controle de fluxo condicional.	Resolução de problemas. Exemplos.	Exercícios práticos
29/06 (T)	Estruturas de repetição. Algoritmos envolvendo repetição.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
01/07 (P)	Exercícios com estruturas condicionais e repetição.	Resolução de problemas. Exemplos.	Exercícios práticos
06/07 (T)	Vetores unidimensionais. Conceito. Operadores.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
07/07 (T)	Prova 1	Avaliação	Avaliação
08/07 (P)	Exercícios com estruturas de repetição e vetores.	Resolução de problemas. Exemplos.	Exercícios práticos
13/07 (T)	Vetores unidimensionais. Algoritmos. Resolução de problemas usando vetores unidimensionais.	Aula expositiva em vídeo. Resolução de problemas. Exemplos.	Exercícios de fixação
15/07 (P)	Exercícios com vetores.	Resolução de problemas. Exemplos.	Exercícios práticos
20/07 (T)	Vetores unidimensionais. Algoritmos. Resolução de problemas usando vetores unidimensionais.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
21/07 (T)	Cadeias de caracteres (strings). Operações. Funções com strings.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
22/07 (P)	Exercícios com strings.	Resolução de problemas. Exemplos.	Exercícios práticos
27/07 (T)	Entrada/saída de dados. Funções de entrada/saída.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
29/07 (P)	Exercícios com entrada/saída de dados.	Resolução de problemas. Exemplos.	Exercícios práticos
03/08 (T)	Matrizes. Conceito. Operações. Algoritmos	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
04/08 (T)	Algoritmos com matrizes. Resolução de problemas envolvendo matrizes.	Aula expositiva em vídeo. Resolução de problemas. Exemplos.	Exercícios de fixação
05/08 (P)	Exercícios com matrizes.	Resolução de problemas. Exemplos.	Exercícios práticos
10/08 (T)	Algoritmos com matrizes. Resolução de problemas envolvendo matrizes.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação

12/08 (P)	Exercícios com matrizes.	Resolução de problemas. Exemplos.	Exercícios práticos
17/08 (T)	Revisão.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
18/08 (T)	Prova 2	Avaliação	Avaliação
19/08 (P)	Exercícios diversos.	Resolução de problemas.	Exercícios de fixação
24/08 (T)	Prova Sub	Avaliação	Avaliação
26/08 (P)	Fechamento de conceitos.	Não há	Não há
30/08 (P)	Prova Rec		

Descrição dos instrumentos e critérios de avaliação qualitativa

Ferramentas: Linguagem de programação Python; Ambientes de desenvolvimento (software livre); Computadores; Navegadores web; Acesso à internet (preferencialmente banda larga).

Critérios de Avaliação:

O cálculo do conceito final (CF) utilizará dois conceitos parciais: conceito do conteúdo teórico (T) e conceito de exercícios práticos (E):

CÁLCULO DO CONTEÚDO TEÓRICO (T):

A avaliação será realizada por meio de duas provas, sendo que cada uma possui os seguintes pesos:

- Prova 1 (P1): peso 1
- Prova 2 (P2): peso 2

Cada prova (P1 e P2) possuirá uma pontuação individual no intervalo de 0-100. A média ponderada (M) das três provas será calculado seguindo a fórmula:

$$M = (P1 + 2 \times P2)/3$$

O conceito final T será obtido a partir do mapeamento da tabela a seguir:

Conceito (T/E)	Médias(M/ME)
A	85 - 100
B	70 - 84
C	55 - 69
D	45 - 54
F	0 - 44

CÁLCULO DAS ATIVIDADES PRÁTICAS (E):

O conceito de atividades práticas (E) será obtido pela média aritmética ponderada dos exercícios desenvolvidos durante o quadrimestre regular. Cada exercício terá uma nota (N) e um peso (W). O cálculo da média de exercícios (ME) será dado pela expressão:

$$ME = (N_1 \cdot W_1 + N_2 \cdot W_2 + N_3 \cdot W_3 + \dots + N_k \cdot W_k) / (W_1 + W_2 + W_3 + \dots + W_k)$$

onde k é o número de exercícios práticos aplicados no quadrimestre.

O conceito E é o mapeamento de ME para o conceito correspondente usando a mesma tabela acima.

CONCEITO FINAL (CF):

O conceito final da disciplina utilizará a seguinte tabela para sua composição:

Conceito Teoria (T)	Conceito Prática (E)	Conceito Final
A	A	A
	B	A
	C	B
	D	B
B	A	B
	B	B
	C	B
	D	C
C	A	B
	B	C
	C	C
	D	C
D	A	C
	B	C
	C	D
	D	D
F	qualquer conceito	F
qualquer conceito	F	F

AVALIAÇÃO SUBSTITUTIVA (SUB):

Em cumprimento à Resolução ConsEPE no. 227/2018, definimos os critérios para avaliação substitutiva.

A prova substitutiva é destinada àqueles que perderam uma das duas provas teóricas do curso, independentemente do motivo.

Obs1: A prova sub será aberta a quem quiser fazer, mas *sua nota irá necessariamente substituir a menor nota do aluno*, mesmo que isso prejudique a nota final.

Obs2: Quem perdeu 2 ou mais provas só poderá substituir mais de uma prova em casos comprovados e previstos na Resolução ConsEPE no. 227/2018.

Para a composição das atividades práticas, os alunos que não puderam entregar seus exercícios por motivos de força maior previstos na Resolução ConsEPE no. 227/2018, poderão requerer novo prazo para entrega dos mesmos ou, se for o caso, nova bateria de exercícios com pontuação equivalente ao período requerido.

AVALIAÇÃO DE RECUPERAÇÃO (REC):

Em cumprimento à Resolução ConsEPE no. 182/2014, todos os alunos que obtiverem conceito final (CF) igual a “D” ou “F” terão direito à realização de avaliação de recuperação, que seguirá os seguintes critérios:

- A composição do conceito final após a recuperação será formada segundo a tabela abaixo:

Conceito final antes da REC	REC	Conceito final do quadrimestre
D	A	C
	B	C
	C	D
	D	D
	F	D*
F	A	C
	B	D
	C	D
	D	F
	F	F

* Para fins de cálculo do conceito final do quadrimestre, garante-se ao aluno o maior conceito entre o obtido antes e após a realização da REC.

PLÁGIOS:

Como a composição da nota final e, conseqüentemente, do conceito final será baseada no desenvolvimento de exercícios práticos de programação, esta disciplina será rigorosa com relação a utilização de códigos de programação plagiados. A fim de preservar o compromisso da universidade com o caráter pedagógico das atividades e o compromisso ético com a propriedade e integridade intelectual, casos suspeitos de plágio serão **severamente** punidos com a **anulação integral de todas as atividades** envolvidas no caso.

REPROVAÇÃO POR AUSÊNCIAS:

Atendendo à Resolução CONSEPE no. 240/2020, que estabelece a autorização de oferta excepcional de quadrimestre suplementar, não serão contabilizadas ausências nesta oferta de disciplina. Sendo assim, não será aplicado o conceito “O” para qualquer aluno matriculado nesta turma.

ATIVIDADES DE APOIO (HORÁRIO DE ATENDIMENTO):

Em cumprimento à Resolução CONSUNI no. 183/2017, os horários de atendimento serão presenciais, indicados a seguir:

- Raphael Yokoingawa de Camargo: Quartas, 14h-16h, sala 266, bloco Delta
- Valerio Batista: Quarta-feira, 14h-15h.

Referências bibliográficas básicas

1. FORBELLONE, André Luiz Villar; EBERSPACHER, Henri Frederico. Lógica de programação: a construção de algoritmos e estruturas de dados. 3 ed. São Paulo: Prentice Hall, 2005. 218 p.
2. SEBESTA, Robert W. Conceitos de linguagens de programação. 5 ed. Porto Alegre: Bookman, 2003. 638 p.
3. Ascensio,A.F.;Campos,E.A.,FundamentosdaProgramaçãodeComputadores, Pearson, 3a edição, 2012.

Referências bibliográficas complementares

1. BOENTE, Alfredo. Aprendendo a programar em Pascal: técnicas de programação. 2003. Rio de Janeiro: Braport, 2003. 266 p.
2. Deitel P.; Deitel, H. "Java - Como Programar" - 8a Ed. São Paulo: Prentice Hall Brasil 2010, I.S.B.N.: 9788576055631 pp 1152.
3. Flanagan, D. "Java, o guia essencial" 5a ed. (série O'Reilly) Bookman Cia Ed 2006 ISBN 8560031073, 1099 pp.
4. SEDGEWICK, Robert; WAYNE, Kevin Daniel. Introduction to programming in Java: an interdisciplinary approach. Boston: Pearson Addison-Wesley, 2007. 723 p
5. Puga, S., Lógica de programação e estruturas de dados com aplicações em Java, Pearson Prentice Hall, 2a edição, 2009