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:	**SA**		
Código da turma:	**DA9BCM0505-15SA**	Turma:	**A9**	Turno:	**Matutino**	Quadrimestre:	**2**	Ano:	**2021**
Docente(s) responsável(is):	**MARCIO K. OIKAWA (T) / MARCIO K. OIKAWA (P)**								

Alocação da turma						
	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
8:00 - 9:00			Teoria (sem. I)	Prática		
9:00 - 10:00			Teoria (sem. I)	Prática		
10:00 - 11:00		Teoria				
11:00 - 12:00		Teoria				
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
25/05 (T)	Apresentação da disciplina; Apresentação do ambiente de programação. Introdução a linguagem de programação utilizada no quadrimestre.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Não há
26/05 (T)	Noções básicas de programação de computadores. Variáveis. Operadores matemáticos. Operadores de atribuição. Precedência.	Aula expositiva em vídeo. Resolução de problemas. Exemplos.	Exercícios de fixação
27/05 (P)	Exercícios de ambientação com programação. Algoritmos sequenciais.	Resolução de problemas. Exemplos.	Exercícios práticos
01/06 (T)	Algoritmos sequenciais com modularização. Módulos. Definição e execução. Parametrização.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
03/06 (P)	Feriado	Recesso	Recesso
08/06 (T)	Estruturas de controle de fluxo. Estruturas condicionais. Operadores comparativos e lógicos.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
09/06 (T)	Encadeamento de estruturas condicionais.	Aula expositiva em vídeo. Resolução de problemas. Exemplos.	Exercícios de fixação
10/06 (P)	Exercícios com modularização e algoritmos sequenciais.	Resolução de problemas. Exemplos.	Exercícios práticos
15/06 (T)	Estruturas condicionais encadeadas.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
17/06 (P)	Exercícios com estruturas condicionais.	Resolução de problemas. Exemplos.	Exercícios práticos
22/06 (T)	Estruturas de repetição. Definição e execução.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
23/06 (T)	Estruturas de repetição baseadas em contagem. Estruturas de repetição condicionais.	Aula expositiva em vídeo. Resolução de problemas. Exemplos.	Exercícios de fixação
24/06 (P)	Exercícios com estruturas de repetição.	Resolução de problemas. Exemplos.	Exercícios práticos
29/06 (T)	Resolução de exercícios envolvendo estruturas de repetição.	Aula expositiva dialogada. Resolução de	Exercícios de fixação

		problemas. Exemplos.	
01/07 (P)	Exercícios com estruturas de repetição.	Resolução de problemas. Exemplos.	Exercícios práticos
06/07 (T)	Vetores. Definição. Operadores e operações úteis. Exemplos.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
07/07 (T)	Resolução de problemas envolvendo vetores.	Aula expositiva em vídeo. Resolução de problemas. Exemplos.	Exercícios de fixação
08/07 (P)	Exercícios com vetores.	Resolução de problemas. Exemplos.	Exercícios práticos
13/07 (T)	Matrizes. Definição de matrizes. Operadores e operações.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
15/07 (P)	Exercícios com matrizes.	Resolução de problemas. Exemplos.	Exercícios práticos
20/07 (T)	Resolução de problemas envolvendo matrizes.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
21/07 (T)	Resolução de problemas envolvendo matrizes.	Aula expositiva em vídeo. Resolução de problemas. Exemplos.	Exercícios de fixação
22/07 (P)	Exercícios com matrizes.	Resolução de problemas. Exemplos.	Exercícios práticos
27/07 (T)	Leitura/Escrita de arquivos.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
29/07 (P)	Exercícios com matrizes.	Resolução de problemas. Exemplos.	Exercícios práticos
03/08 (T)	Revisão.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
04/08 (T)	Revisão.	Aula expositiva dialogada. Resolução de problemas. Exemplos.	Exercícios de fixação
05/08 (P)	Exercícios diversos.	Resolução de problemas.	Exercícios de fixação
10/08 (T)	Fechamento de conceitos.	Não há	Não há
12/08 (P)	Lançamento de conceitos.	Não há	Não há
16/08			

Descrição dos instrumentos e critérios de avaliação qualitativa

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

Critérios de Avaliação: Os alunos são avaliados a partir de seu desenvolvimento nos tópicos cobertos pela ementa da disciplina. Para isso, serão utilizados exercícios práticos que devem ser desenvolvidos semanalmente e avaliados de forma semiautomática e online. Cada exercício possui um peso (P) e uma pontuação (N) que variam de 0 a 100. O conceito final da disciplina será calculado a partir das notas obtidas pelas entregas dos exercícios, obedecendo ao critério citado a seguir.

CÁLCULO DO CONCEITO FINAL:
O conceito final (CF) será calculado a partir do seguinte cálculo:

$CF = (N_1*P_1 + N_2*P_2 + N_3*P_3 + ... + N_k*P_k)/(P_1+P_2+P_3+...+P_k)$

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

A atribuição do conceito será dada conforme a tabela a seguir.

Conceito	CF
A	90 – 100
B	75 – 89
C	60 – 74
D	50 – 59
F	0 – 49

AVALIAÇÃO SUBSTITUTIVA (SUB):
Em cumprimento à Resolução ConsEPE no. 227/2018, definimos os critérios para avaliação substitutiva.
Como a composição de conceito prevê somente a utilização de exercícios práticos, não havendo provas escritas ou similares, os alunos que forem impedidos de desenvolver e entregar exercícios práticos nos casos previstos na resolução citada terão direito a prazo adicional para entrega dos mesmos.

AVALIAÇÃO DE RECUPERAÇÃO (REC):

Em cumprimento à Resolução ConsEPE no. 182/2014, todos os alunos que obtiverem conceito final 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, consequentemente, 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 e CONSEPE no. 240/2020, os horários de atendimento não serão presenciais, de forma que os alunos interessados poderão agendar horários de atendimento diretamente com o professor da disciplina por e-mail ou algum dos mecanismos de comunicação utilizados na condução das atividades.

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