

Unidade curricular

# Lógica de Programação e Algoritmos

**Unidade curricular (UC):** Módulo Básico - Lógica de Programação e Algoritmos**Carga horária da UC:** 75 horas**Nº de aulas:** 100 aulas**Carga horária prevista para o desenvolvimento da Situação de Aprendizagem:** 75 horas**Objetivo da UC:** proporcionar a aquisição de capacidades básicas e socioemocionais que permitem desenvolver algoritmos, por meio de lógica de programação e versionamento, para resolução de problemas.

## CrITÉRIOS de avaliação

INSTRUMENTO DE REGISTRO			
Nome do aluno:		Turma:	
Capacidades básicas ou técnicas e socioemocionais	CrITÉRIOS de Avaliação	Resultado	
		Aluno	Docente
1. Identificar a sequência lógica de passos em um algoritmo	<b>O aluno descreveu os passos necessários para solução do problema proposto?</b>		
	<i>O aluno disponibilizou os passos detectados em ordem sequencial?</i>		
2. Utilizar lógica de programação para a resolução de problemas	<b>O aluno demonstrou compreensão sobre o problema proposto analisando e propondo os requisitos e restrições para a solução esperada?</b>		
	<i>O aluno efetuou a decomposição do problema proposto em partes menores facilitando a compreensão e a solução?</i>		
	<b>O aluno aplicou as estruturas de condição para controlar o fluxo do programa de acordo com as condições necessárias para a resolução do problema proposto?</b>		
	<i>O aluno escolheu os tipo de variáveis apropriadas para o programa?</i>		
3. Demonstrar pensamento analítico	<b>O aluno utilizou as informações apresentadas no enunciado do problema proposto para compreender o cenário e fazer inferência a partir desses dados?</b>		
4. Demonstrar inteligência emocional	<i>O aluno apresentou comportamento de tranquilidade e calma diante dos desafios apresentados durante a solução do problema proposto considerando a complexidade e cumprimento de prazos?</i>		
	<b>O aluno soube lidar com os fracassos e adversidades que surgiram durante a solução do problema de modo a interpretá-los como uma oportunidade de aprendizagem e resiliência?</b>		
5. Representar algoritmos por meio de fluxogramas, seguindo as	<b>O aluno definiu os passos de execução para o problema proposto?</b>		

INSTRUMENTO DE REGISTRO			
convenções de símbolos e conexões	O aluno representou os passos de execução do programa em um diagrama de fluxo?		
	O aluno utilizou as representações de cada operação do fluxograma de forma correta?		
6. Utilizar variáveis para armazenar valores durante a execução de um programa	O aluno criou variáveis para armazenar os valores utilizados pelo programa?		
	O aluno escolheu nomes de variáveis de acordo com as boas práticas do clean code?		
	O aluno identificou os diferentes tipos de variáveis utilizadas no programa?		
7. Utilizar operadores aritméticos para realizar cálculos em expressões numéricas	O aluno identificou os diferentes símbolos de operadores matemáticos para realizar as operações aritméticas do programa?		
	O aluno empregou as regras de precedência durante a implementação de expressões numéricas no programa?		
8. Utilizar tomada de decisão para elaboração do algoritmo	O aluno escreveu as condições necessárias para a tomada de decisão?		
	O aluno codificou as instruções para tomada de decisão de acordo com a estrutura de decisão selecionada?		
	O aluno utilizou as saídas alternativas da instrução condicional de acordo com a sintaxe do método escolhido?		
9. Utilizar estruturas condicionais para executar instruções com base em uma condição	O aluno utilizou as condições necessárias para a tomada de decisão?		
	O aluno escreveu a instrução para tomada de decisão de acordo com a sintaxe da estrutura condicional escolhida?		
	O aluno utilizou as saídas alternativas da instrução condicional de acordo com a sintaxe da estrutura de decisão selecionada?		
10. Demonstrar autogestão	O aluno cumpriu os prazos para entrega das atividades propostas?		
	O aluno demonstrou organização pessoal incluindo o gerenciamento dos seus pertences, anotações do conteúdo?		
11. Criar estruturas de repetição para executar um conjunto de instruções várias vezes	O aluno escreveu as condições necessárias para a ocorrência do laço de repetição?		
	O aluno codificou as instruções de laço de repetição de acordo com a sintaxe da estrutura de controle escolhida?		
	O aluno aplicou diferentes estruturas de controle ao código escrito?		
12. Aplicar operadores lógicos para avaliar e combinar condições booleanas	O aluno utilizou os operadores lógicos na construção de operações para tomadas de decisão no programa?		
	O aluno combinou operadores lógicos para flexibilizar a tomada de decisão?		
13. Demonstrar autonomia	O aluno concluiu as tarefas de forma independente?		

INSTRUMENTO DE REGISTRO			
	<i>O aluno buscou oportunidades de aprendizado além daquelas apresentadas em aula para resolver o problema proposto?</i>		
	<i>O aluno demonstrou criatividade e inovação na execução das atividades adicionando funcionalidades ou utilização de recursos que excederam as exigências para a solução do problema proposto?</i>		
14. Utilizar vetores e matrizes na elaboração do programa	<b>O aluno declarou o vetor e/ou matriz de acordo com a linguagem utilizada?</b>		
	<b>O aluno acessou o elemento do vetor e/ou matriz de acordo com a sintaxe da linguagem utilizada?</b>		
	<i>O aluno iterou nos elementos do vetor e/ou matriz de acordo com a sintaxe da linguagem de programação utilizada?</i>		
	<i>O aluno manipulou os dados do vetor e/ou matriz utilizando a sintaxe correta?</i>		
15. Aplicar técnicas de código limpo (clean code)	<b>O aluno escolheu nomes de variáveis e funções de forma a revelar o seu propósito?</b>		
	<i>O aluno evitou o uso de abreviações ao nomear variáveis e funções?</i>		
	<i>O aluno evitou comentários óbvios ou redundantes no programa?</i>		
	<i>O aluno evitou o uso de código duplicado através da aplicação da responsabilidade única?</i>		
16. Manipular os diferentes tipos de dados na elaboração de programas	<b>O aluno criou variáveis de diferentes tipos na construção do programa?</b>		
	<i>O aluno efetuou o casting de tipos de dados entre diferentes tipos</i>		
17. Utilizar o ambiente integrado de desenvolvimento (IDE)	<b>O aluno utilizou uma IDE no desenvolvimento da aplicação?</b>		
	<b>O aluno organizou o projeto de acordo com as recomendações de design pattern da linguagem de programação utilizada?</b>		
	<i>O aluno utilizou os recursos de auto formatação de código disponibilizado pela IDE selecionada?</i>		
18. Demonstrar autogestão	<b>O aluno cumpriu os prazos para entrega das atividades propostas?</b>		
	<i>O aluno demonstrou organização pessoal incluindo o gerenciamento dos seus pertences, anotações do conteúdo?</i>		
19. Criar repositórios Git locais e remotos para controle de versionamento	<i>O aluno criou o repositório local?</i>		
	<i>O aluno adicionou os arquivos a lista de monitoramento do git?</i>		
	<i>O aluno efetuou os commits adicionando uma mensagem que descreve as alterações feitas?</i>		
	<i>O aluno adicionou o arquivo ".gitignore" com a relação de pastas e arquivos que não devem ser monitorados pelo git?</i>		
	<i>O aluno alterou o nome da branch principal para main?</i>		

## INSTRUMENTO DE REGISTRO

	O aluno criou um repositório no <i>GitHub</i> ?		
	O aluno sincronizou o repositório local com o <i>GitHub</i> ?		
	O aluno criou um arquivo <i>README.MD</i> para documentar o projeto disponibilizado no <i>GitHub</i> ?		

## LEGENDA:

Critérios críticos: **Vermelho ou Negrito**

Critérios desejáveis: Preto ou Itálico

## TABELA DE NÍVEIS DE DESEMPENHO

Capacidades básicas/específicas e socioemocionais	Nível de desempenho	Conversão em notas
Atingiu todos os críticos (30) e todos os desejáveis (26)	10	100
Atingiu todos os críticos e 20 desejáveis	9	90
Atingiu todos os críticos e 15 desejáveis	8	80
Atingiu todos os críticos e 10 desejáveis	7	70
Atingiu todos os críticos e 02 desejáveis	6	60
Atingiu todos os críticos	5	50
Atingiu menos de 30 críticos	4	40
Atingiu menos de 26 críticos	3	30
Atingiu menos de 16 críticos	2	20
Atingiu menos de 6 críticos	1	10