

Desenvolvimento de Back-End

Anthony Freitas



Programador Back-End

Instrutor: Anthony Samuel Sobral de Freitas

anthony.freitas@docente.senai.br

- Engenheiro de Software
- Quality Assurance Tester
- Designer de UI/UX

bit.ly/curriculo anthony







Programador Back-End

Módulos	Unidades Curriculares	Aulas	Carga Horária	Carga Horária do Módulo
Módulo Básico	Lógica de Programação	7	28	- 60
	Banco de Dados	8	32	
Módulo Específico Profissional	Modelagem de Sistemas	4	16	120
	Programação Orientada a Objetos	7	28	
	Desenvolvimento de API Restful	14	56	
	Integração de Competências Profissionais	5	20	
Carga Horária Total		45		180



Lógica de Programação:

O que é Programação?

• A programação é o processo de instruir um computador para realizar tarefas específicas. Isso é feito através de um conjunto de instruções precisas, conhecidas como código.





Por que programar é importante?

- Automatização de Tarefas: A programação permite automatizar processos repetitivos, economizando tempo e minimizando erros.
- Solução de Problemas: Permite a criação de soluções para uma ampla variedade de problemas, desde simples cálculos até complexos sistemas.
- Inovação e Tecnologia: A programação é a base para o desenvolvimento de aplicativos, software e tecnologias que impulsionam a inovação em diversas áreas.
- Empregabilidade: Há uma demanda crescente por profissionais de programação em diversas indústrias, tornando a habilidade de programar uma vantagem competitiva no mercado de trabalho.



O que são algoritmos?

Definição de Algoritmos

- Um algoritmo é um conjunto de instruções ou regras bem definidas que guiam a execução de uma tarefa ou resolução de um problema.
- Algoritmos são essenciais na programação, pois fornecem o plano de ação para realizar uma determinada operação.

Características dos Algoritmos

- **Precisão:** Cada passo deve ser definido claramente, sem ambiguidades.
- Finitude: O algoritmo deve terminar em um número finito de etapas.
- Efetividade: Cada instrução deve ser clara e executável.



A Importância de um Algoritmo Eficiente:

Economia de Recursos

 Algoritmos eficientes economizam tempo e recursos computacionais.

Resolução de Problemas Complexos

• Permitem abordar problemas complexos de forma organizada e escalável.

Melhoria da Experiência do Usuário

 Em aplicações e sistemas, algoritmos eficientes garantem uma resposta rápida e uma experiência mais fluida para o usuário final.

Benefícios para a Indústria e Inovação

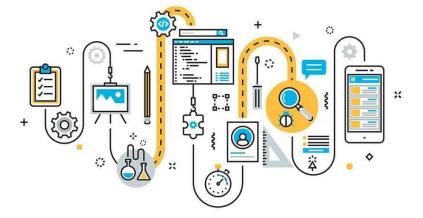
• Empresas e indústrias inovadoras dependem de algoritmos eficientes para otimizar processos e criar novas soluções.



Algoritmos

Tipos de Algoritmos

- Descrição narrativa utiliza linguagem natural para especificar os passos para a realização das tarefas ou solução de um problema.
- Fluxograma é uma forma universal de representação, pois se utiliza de figuras geométricas para ilustrar os passos a serem seguidos para a resolução dos problemas.
- Pseudocódigo utiliza linguagem estruturada e se assemelha na forma de um programa escrito em linguagem de programação estruturada, também conhecido como português estruturado ou Portugol, muito usado no meio computacional.





Descrição Narrativa



Utiliza linguagem natural para especificar os passos para a realização das tarefas ou solução de um problema.



Algoritmo narrativo sequencial.

Problema observado: Preparar um bolo (Exemplo 01)

inicio

- Bata quatro claras em neves.
- Adicione duas xícaras de açúcar.
- Adicione duas xícaras de farinha de trigo, quatro, gemas, uma colher de fermento e duas colheres de chocolate.
- Bata por três minutos.
- Coloque em uma assadeira com margarina e farinha de trigo.
- Coloque o bolo no forno para assar durante vinte minutos.
- Retire o bolo do forno.

Problema observado: Trocar uma lâmpada (Exemplo 02)

Inicio

- · Pegue uma escada;
- Posicione a escada embaixo da lâmpada;
- Pegue uma lâmpada nova;
- Suba na escada;
- · Retire a lâmpada velha;
- Coloque a lâmpada nova.

Fim

Algoritmo Narrativo com decisão (Condicional):

- Podemos ter sequências de tarefas onde alguns passos podem ou não serem executados dependendo de algumas condições que o processo pode nos oferecer
- Para esses casos podemos usar os algoritmos com decisões ou condicionais
- No exemplo anterior fizemos um algoritmo para trocar uma lâmpada, mas não verificamos se a lâmpada precisaria ser trocada.
- Segue um exemplo para essa situação usando uma condição



Descrição narrativa usando condicional (Exemplo 01)

Inicio

- Ligue o interruptor
- Se a lâmpada não acender:
 - Pegue uma escada;
 - Posicione a escada embaixo da lâmpada;
 - Pegue uma lâmpada nova;
 - Suba na escada;
 - Retire a lâmpada velha;
 - Coloque a lâmpada nova

Fim