

# Guia de Estudos – Algoritmos I

Resumo das 40 aulas, planos de preparação e acompanhamento, e referências para aprofundamento.

## Como usar este guia

Utilize este material como referência semanal. Cada aula inclui objetivos conceituais e práticos, atividades de reforço e conexões com avaliações. As seções de preparação, condução e revisão oferecem checklists que ajudam a manter a disciplina. Combine o guia com os exercícios oficiais da disciplina e as rubricas publicadas no ambiente virtual.

## Resumo das 40 aulas

**Aula 01 – Boas-vindas e contrato pedagógico:** Apresentar a disciplina, ferramentas de apoio e expectativas de participação. Configurar ambiente local e revisar noções básicas de lógica.

**Aula 02 – Pensamento algorítmico:** Discutir decomposição de problemas e representação por fluxogramas e pseudocódigo. Exercitar com situações do cotidiano.

**Aula 03 – Variáveis e tipos primitivos:** Introduzir declaração, atribuição e regras de tipagem. Praticar com leitura de dados e saída formatada.

**Aula 04 – Operadores aritméticos e precedência:** Explorar ordem de avaliação, operadores módulo e potência. Resolver exercícios de cálculos compostos.

**Aula 05 – Operadores relacionais e lógicos:** Definir comparações, tabelas verdade e curto-circuito. Aplicar em diagnósticos de regras de negócio simples.

**Aula 06 – Estruturas condicionais simples:** Implementar decisões com if/else. Construir pequenos programas de tomada de decisão.

**Aula 07 – Condicionais aninhadas e múltiplas:** Trabalhar cascatas, else-if e switch/case. Reestruturar decisões complexas em blocos legíveis.

**Aula 08 – Validação de entradas:** Projetar verificações de dados usando condicionais. Introduzir mensagens de erro significativas e repetição de leitura.

**Aula 09 – Laços while:** Apresentar repetição com condição de parada. Aplicar a contadores e acumuladores.

**Aula 10 – Laços do-while:** Comparar testes no início e no final. Ajustar algoritmos que exigem execução mínima garantida.

**Aula 11 – Laços for:** Explorar iteração controlada por contador. Destacar uso em sequências e progressões.

**Aula 12 – Controle de laços (break/continue):** Discutir impacto na legibilidade e estratégias alternativas. Aplicar a buscas e filtros.

**Aula 13 – Aninhamento de laços:** Resolver problemas com matrizes e combinações. Otimizar variáveis de controle.

**Aula 14 – Vetores – conceitos e declaração:** Introduzir vetores, índices e limites. Tratar leitura sequencial de dados em memória.

**Aula 15 – Operações básicas com vetores:** Implementar buscas, contagem e agregações. Discutir complexidade temporal  $O(n)$ .

**Aula 16 – Ordenação por seleção:** Estudar algoritmo selection sort. Analisar trocas, complexidade e estabilidade.

**Aula 17 – Ordenação por inserção:** Comparar insertion sort com selection sort. Analisar uso em subconjuntos quase ordenados.

**Aula 18 – Busca linear e binária:** Avaliar quando usar cada abordagem. Praticar com listas ordenadas e não ordenadas.

**Aula 19 – Introdução a matrizes:** Declarar e percorrer matrizes bidimensionais. Aplicar em tabelas e mapas.

**Aula 20 – Operações com matrizes:** Somar, transpor e multiplicar matrizes. Utilizar laços aninhados de forma eficiente.

**Aula 21 – Funções – motivação e sintaxe:** Discutir modularização e reuso. Definir funções sem retorno e com retorno.

**Aula 22 – Parâmetros e escopo:** Explorar passagem por valor e por referência (quando aplicável). Explicar vida útil das variáveis.

**Aula 23 – Bibliotecas padrão:** Apresentar funções utilitárias e organização de headers. Incentivar leitura de documentação.

**Aula 24 – Boas práticas com funções:** Cobrir nomenclatura, coesão e efeitos colaterais. Introduzir testes unitários simples.

**Aula 25 – Strings e manipulação básica:** Trabalhar leitura, concatenação e comparação. Demonstrar uso de funções da biblioteca padrão.

**Aula 26 – Arquivos – entrada e saída:** Configurar leitura e escrita em arquivos texto. Implementar validação e fechamento seguro.

**Aula 27 – Estruturas de dados compostas:** Definir structs e registros. Relacionar com vetores e matrizes para modelos de dados complexos.

**Aula 28 – Projeto modular – estudo de caso:** Planejar módulos, interfaces e responsabilidades. Criar diagrama de componentes simples.

**Aula 29 – Depuração e tratamento de erros:** Utilizar ferramentas de debug, breakpoints e logs. Revisar mensagens claras ao usuário.

**Aula 30 – Análise de complexidade:** Introduzir notação Big-O. Estimar custos de algoritmos vistos até aqui.

**Aula 31 – Recursão – conceitos básicos:** Explicar casos base e passo recursivo. Resolver exemplos clássicos como fatorial e Fibonacci.

**Aula 32 – Recursão em estruturas de dados:** Aplicar em busca binária recursiva e percursos em árvores abstratas.

**Aula 33 – Estratégias de teste:** Planejar casos de teste e critérios de cobertura. Documentar planos de teste.

**Aula 34 – Projeto integrador – briefing:** Apresentar requisitos e critérios de avaliação do projeto final. Dividir equipes e tarefas.

**Aula 35 – Projeto integrador – iteração 1:** Focar em estrutura de dados e interface de entrada. Entregar protótipo funcional parcial.

**Aula 36 – Projeto integrador – iteração 2:** Realizar refino, otimizações e validação com usuários simulados.

**Aula 37 – Revisão geral de algoritmos:** Relembrar principais padrões de decisão, repetição e coleções. Resolver listas de exercícios síntese.

**Aula 38 – Simulado para avaliação final:** Aplicar prova modelo com feedback imediato. Orientar estratégias de gerenciamento de tempo.

**Aula 39 – Oficina de dúvidas:** Atender dificuldades específicas e revisar tópicos críticos conforme demanda.

**Aula 40 – Preparação para showcase:** Orientar apresentações, storytelling técnico e critérios de avaliação externa.

**Aula 41 – Showcase e feedback final:** Apresentar projetos, coletar feedback de pares e professores, e discutir próximos passos na jornada de programação.

## Roteiro de Preparação (antes da aula)

1. Leia o resumo da aula anterior e identifique dúvidas para levar ao encontro seguinte.

2. Revise o roteiro pré-aula no ambiente virtual (slides, vídeos curtos e checklist).

3. Configure ou atualize o ambiente de desenvolvimento (IDE, compilador e extensões).

4. Resolva pelo menos dois exercícios de aquecimento indicados na trilha de prática.

5. Note objetivos pessoais para a aula (conceitos que deseja dominar ou lacunas a preencher).

## Roteiro Durante a Aula

1. Compare exemplos do professor com suas anotações e destaque diferenças importantes.

2. Faça perguntas assim que identificar lacunas conceituais ou dificuldades nos exercícios guiados.

3. Participe das atividades colaborativas anotando decisões de design e justificativas.

4. Capture evidências do raciocínio (pseudocódigo, prints de execução, resultados de testes).

5. No final, revise objetivos definidos na preparação e confirme se foram atendidos.

## Roteiro Pós-aula

1. Finalize exercícios complementares sugeridos e envie para correção automática quando disponível.

2. Atualize seu diário de aprendizagem destacando conquistas, dificuldades e próximos passos.

3. Revise código produzido, refatore trechos pouco legíveis e documente decisões importantes.

4. Compartilhe dúvidas no fórum ou grupos de estudo e acompanhe feedbacks recebidos.

5. Planeje tempo para revisar os tópicos antes da avaliação correspondente (quiz, laboratório ou projeto).

# Acompanhamento de Avaliações

Utilize a tabela a seguir para monitorar entregas práticas e checkpoints do projeto integrador. Atualize semanalmente com status e evidências de conclusão.

- 1 **Laboratórios práticos (L1–L6):** Aplicação direta de loops, vetores e matrizes. Valorize clareza de código e testes.
- 2 **Quizzes formativos:** Consolide teoria de forma rápida; use feedback imediato para direcionar estudos.
- 3 **Projeto integrador:** Iterações 1 e 2 focam em modularização, persistência de dados e testes.
- 4 **Showcase final:** Demonstração do projeto para banca convidada com foco em comunicação técnica.

## Indicadores de Progresso

Taxa de exercícios concluídos por módulo (meta:  $\geq 80\%$ ).

Resultados de quizzes (meta:  $\geq 70\%$  antes da prova final).

Cobertura de testes no projeto (meta: casos para fluxos principais e cenários de erro críticos).

Feedback qualitativo do professor e dos pares após atividades síncronas.

# Referências confiáveis

☪ormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Algoritmos: Teoria e Prática*. MIT Press.

ℤelle, J. (2017). *Python Programming: An Introduction to Computer Science*. Franklin, Beedle & Associates.

ℑedgewick, R., & Wayne, K. (2023). *Algorithms* (4th ed.). Addison-Wesley.

MIT OpenCourseWare – *Introduction to Computer Science and Programming* (6.0001).  
<https://ocw.mit.edu>

Courseera – *Programming Foundations with JavaScript, HTML and CSS* (Duke University).

Harvard CS50x – *Introduction to Computer Science*. <https://cs50.harvard.edu>

Recomenda-se validar a aderência do conteúdo às políticas institucionais e atualizar links semestrais.  
Inclua novas fontes conforme evolução da disciplina.

# Plano Semanal de Revisão

A tabela a seguir sugere distribuição do tempo fora da aula. Adapte conforme disponibilidade e dificuldade pessoal.

**S**egunda-feira: Revisar anotações da aula anterior e atualizar diário de aprendizagem (30 minutos).

**T**erça-feira: Resolver exercícios curtos focados em conceitos novos (45 minutos).

**Q**uarta-feira: Trabalhar no projeto integrador ou laboratório vigente (60 minutos).

**Q**uinta-feira: Participar de grupo de estudos ou fórum para discutir dúvidas (30 minutos).

**S**exta-feira: Realizar quiz de autoavaliação ou simulado curto (30 minutos).

**F**im de semana: Consolidar código, escrever testes adicionais e planejar semana seguinte (60 minutos).

## Checklist rápido

**E**stou acompanhando o cronograma de 40 aulas sem lacunas?

**T**enho evidências de aprendizagem (código versionado, relatórios, reflexões)?

**S**ei onde encontrar ajuda (monitores, fóruns, plantões)?

**A**gendei revisões antes de cada avaliação formal?

Última atualização: Janeiro 2025 • Responsável: Equipe EDU