
INTERFATECS

Maratona de Programação

Prof. Alex Marino

alex.marino@fatecourinhos.edu.br

16 de agosto de 2025

Sumário

- 1 Visão Geral
- 2 Como Funciona
- 3 Problemas e Habilidades
- 4 Estratégia
- 5 Preparação
- 6 Encerramento

Visão Geral

O que é a Maratona de Programação?

- Competição de algoritmos e programação em **equipes de 3**.
- Tempo típico: **5 horas** para resolver **8–13 problemas**.
- Baseada em competições como *ICPC* e a *Maratona SBC*.
- Foco: **raciocínio lógico**, **estruturas de dados**, **trabalho em equipe** e **gestão do tempo**.

- ICPC surge nos anos 1970 (meados da década) em ambiente universitário.
- No Brasil, a Maratona de Programação da **SBC** acontece desde a década de 1990.
- Etapas: *Seletivas locais* → *Regionais* → *Nacional* → *Mundial (ICPC)*.

Como Funciona

Formato e regras essenciais

- **Equipe:** 3 competidores, 1 computador.
- **Problemas:** múltiplos níveis (fáceis a avançados).
- **Linguagens:** C/C++, Java, Python (varia por regulamento).
- **Entradas/Saídas:** padrão (stdin/stdout); leitura robusta e *parsing* cuidadoso.
- **Comunicação:** apenas entre membros da equipe (sem internet, exceto enunciados/FAQ do juiz).

Pontuação e desempate

- **Classificação:** mais problemas resolvidos.
- **Desempate:** menor tempo total (tempo até a submissão aceita) + **penalidades** por erros prévios.

Submissão aceita	Soma tempo desde o início + 20 min por cada WA anterior
Submissão rejeitada	Não soma tempo (apenas penaliza se houver aceitação futura)

Dica: evite **chutar**; teste localmente e envie com confiança.

Problemas e Habilidades

Tipos comuns de problemas

- **Estruturas de Dados:** filas, pilhas, heaps, mapas.
- **Grafos:** BFS/DFS, Dijkstra, MST, fluxo.
- **DP:** knapsack, caminhos, subsequências.
- **Matemática:** número primo, congruências, combinatória.
- **Geometria:** interseções, convex hull, ângulos.
- **Strings:** prefixo, KMP, Z, suffix array/trie.

- **Modelagem:** transformar texto do enunciado em *abstrações* computacionais.
- **Complexidade:** estimar $O(n)$, $O(n \log n)$, $O(n^2)$ para caber no tempo.
- **Implementação limpa:** I/O rápido, esquemas de testes e *debug* sob pressão.
- **Trabalho em equipe:** divisão de papéis, comunicação e *handoff* de código.

Estratégia

Estratégia de prova (papéis e fluxo)

- **Papéis sugeridos:** *Leitor/Triador, Implementador, Validador/Testador.*
- **Primeiros 15-20 min:** **Triagem**; marcar fáceis, médios e difíceis.
- **Fila de implementação:** atacar problemas fáceis primeiro para entrar no placar.
- **Teste local:** criar casos próprios e *corner cases*.
- **Controle de tempo:** *timebox* por problema; evitar afundar em um único item.

Erros comuns (e como evitar)

- **Off-by-one** e estouro de índices \Rightarrow asserts e testes mínimos.
- **Overflow** (int vs. long long) \Rightarrow tipos corretos, checar limites.
- **Parsing frágil** \Rightarrow tratar múltiplos espaços/linhas; cuidado com `scanf/cin`.
- **Falta de casos extremos** \Rightarrow construa *stress tests* simples.
- **Complexidade alta** \Rightarrow otimizar estrutura/algoritmo antes de micro-otimizar código.

Preparação

Roadmap de preparação (6–8 semanas)

- 1 Fundamentos: I/O, ordenação, busca binária, prefix sums.
- 2 Grafos: BFS/DFS, árvores, caminhos mínimos.
- 3 DP clássica: subsequências, knapsack, caminhos em grade.
- 4 Matemática: primalidade, fatoração leve, combinatória mod.
- 5 Strings: prefix function, KMP, Z-function.
- 6 Geometria básica: orientação, interseção de segmentos.
- 7 Semana final: **simulados cronometrados** + revisão de *templates*.

- **Templates** por linguagem (I/O rápido, estruturas padrão).
- **Snippets** de DS/algoritmos testados (Union-Find, Dijkstra, etc.).
- **Checklist de envio:** *leu tudo? tratou casos limite? limpa debug?*
- **Plataformas de treino:** Beecrowd, Codeforces, AtCoder, CodeChef, LeetCode (seleção).

Encerramento

Próximos passos

- Forme equipes e definam **rotina de treinos semanais**.
- Montem um **repositório de templates e guia de estilo**.
- Rodem **simulados cronometrados** (2–3 até a competição).
- Ajustem papéis e comunicação durante os treinos.

Competidores, Coaches e Juízes

- São os participantes diretos da competição
- Trabalham individualmente ou em equipe
- Funções principais:
 - Ler e interpretar problemas rapidamente
 - Desenvolver soluções corretas e eficientes
 - Implementar código em linguagens permitidas
 - Submeter e corrigir soluções dentro do tempo limite

O Papel do Coach na Maratona de Programação

Orientador, Motivador e Estrategista

- Organiza e treina a equipe
- Realiza simulados e maratonas internas
- Ensina algoritmos, estruturas de dados e estratégias
- Ajuda na divisão de papéis dentro da equipe
- Motiva e prepara psicologicamente os competidores

Durante a Competição

- Não pode ajudar diretamente os competidores
- Não tem acesso aos problemas nem às soluções
- Acompanha de fora, respeitando as regras de imparcialidade
- Atua como apoio moral e organizacional

Funções Indiretas Durante a Prova

- Estimula o espírito esportivo da equipe
- Ajuda a manter os competidores focados e confiantes
- Observa pontos de melhoria para treinamentos futuros
- Garante que a equipe siga o regulamento e o código de conduta

Após a Competição

- Analisa o desempenho da equipe
- Discute erros e acertos
- Planeja treinamentos futuros
- Incentiva os alunos a continuar participando

Síntese do Papel do Coach

Em resumo

O coach é responsável pela **formação técnica, motivação psicológica e organização estratégica** da equipe. Durante a competição, sua atuação é indireta, mas essencial para o sucesso dos competidores.

- Organizadores e avaliadores oficiais da maratona
- Funções principais:
 - Elaborar e revisar os problemas da competição
 - Definir os casos de teste e critérios de avaliação
 - Monitorar o andamento da prova
 - Julgar submissões de forma automática e justa
 - Resolver dúvidas e manter a integridade da competição
- Garantem a imparcialidade e o espírito esportivo

- **Competidores:** foco em resolver problemas
- **Coaches:** foco em preparar e apoiar as equipas
- **Juízes:** foco em garantir a justiça e o regulamento

"Sem juízes não há competição justa; sem coaches não há preparação sólida; sem competidores não há maratona."

FIM