

# Introdução à Maratona de Programação

v. 1.8

Fernando Kiotheka e Raul Almeida

2023



## 1 Introdução

Bem-vindo ao grupo de estudos para a maratona de programação da Universidade Federal do Paraná, os **Competidores(as) Pirados em Maratona**, o Capimara! Aqui apresentamos uma orientação inicial do que é programação competitiva, como funcionam as principais competições e como nos preparamos para este grande desafio.

## 1.1 O que é isso?

A maratona de programação é uma competição de *programação competitiva*. A programação competitiva é um esporte mental (como por exemplo, o xadrez) onde os participantes precisam criar programas de acordo com uma especificação, isto é, programas que resolvem problemas computacionais.

Uma especificação de problema descreve o que é uma saída correta esperada para uma entrada e também descreve o formato da entrada. O seu objetivo é criar uma solução, um programa que produz as saídas corretas em um tempo adequado, resolvendo o problema proposto.

Em um contexto de competição, várias pessoas obtém o mesmo conjunto de problemas, e vence quem resolver mais problemas. Em caso de empate, vence quem resolver os problemas em menos tempo.

Geralmente, as competições são feitas em equipes. O time pode ser mais eficiente em resolver problemas por meio da troca de ideias e da distribuição de tarefas entre os integrantes.

As competições mais importantes são:

- **Maratona de Programação da SBC/ICPC:** Estudantes de ensino superior de *qualquer* curso.
- **Olimpíada Brasileira de Informática:** Estudantes desde o ensino fundamental até o primeiro ano de graduação.

A competição tem sobretudo, um limite de idade, mas depois é possível participar da Maratona da SBC como café-com-leite, ou participar de competições *online* promovidas por vários *sites*. Porém, como este livreto é para alunos de graduação, o nosso foco vai ser nessas duas competições.

## 1.2 “A” Maratona de Programação (da SBC, ICPC)

A principal competição quando se é estudante de graduação é a **Maratona de Programação da SBC** (Sociedade Brasileira de Computação). Nela, times de **exatos três estudantes cada** representam suas universidades em uma competição regional e outra nacional. As equipe classificadas participam de uma final latino-americana, e na fase seguinte, na final mundial em eventos do **ICPC (International Collegiate Programming Contest)**, organizados pela **ACM (Association for Computer Machinery)**, que sempre acontecem em um país diferente. Cada equipe dispõe de um computador (sem conexão) e qualquer material impresso que levar para fazer a prova de 5 horas.

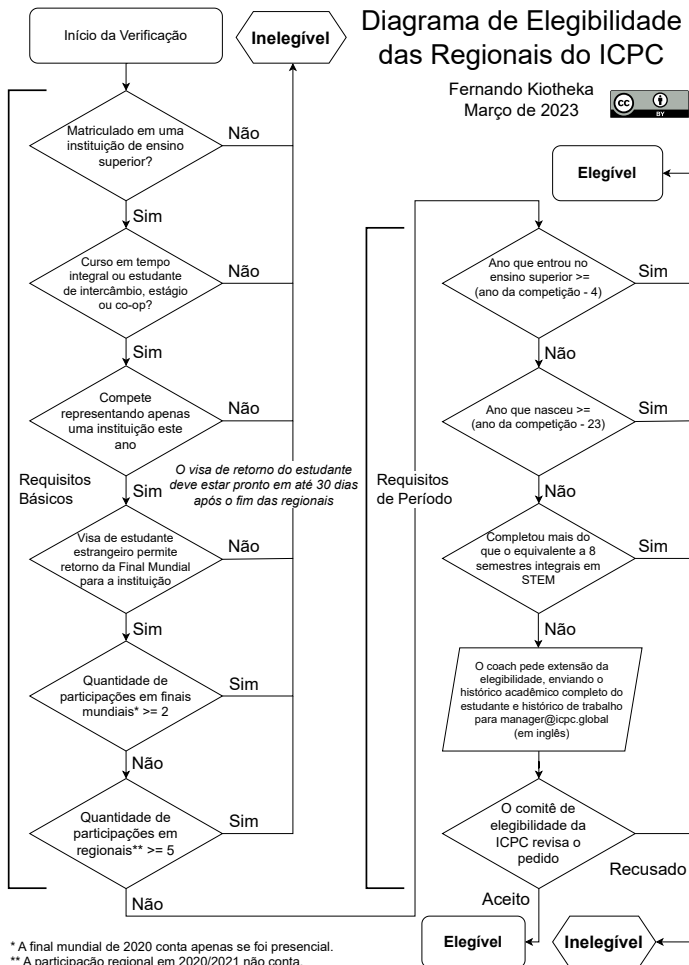


**acm** International Collegiate  
Programming Contest

O critério de elegibilidade é descrito pelo fluxograma:

## Diagrama de Elegibilidade das Regionais do ICPC

Fernando Kiotheka  
Março de 2023



\* A final mundial de 2020 conta apenas se foi presencial.

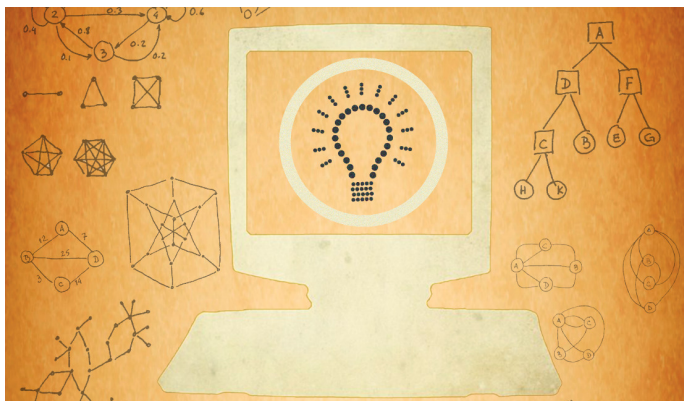
\*\* A participação regional em 2020/2021 não conta.

### 1.3 Olimpíada Brasileira de Informática

A Olimpíada Brasileira de Informática é outra maratona, organizada pela UNICAMP com o objetivo de despertar o interesse no estudo da computação. Ela é aplicada da mesma forma que a Olimpíada de Matemática, isto é, no ensino fundamental e no ensino médio, mas também no ensino superior.

A modalidade que você calouro poderá participar é da nível Sênior na Modalidade Programação. Ela só pode ser feita no primeiro ano de graduação, e diferente das outras maratonas, a a sua pontuação fica disponível apenas alguns dias depois.

É necessário um conhecimento mínimo de programação e para se classificar, e até certo domínio de certas estruturas de dados e algoritmos. Pode ser usado Pascal, C, C++, Java, JavaScript e Python como linguagens de programação e a prova é feita individualmente, sem consulta a materiais externos.



## 2 Motivação

Existem muitos motivos pelos quais você pode ter se interessado pela maratona, e muitos outros que movem outros competidores, mas em uma coisa todos concordarão: participar da maratona de programação é tudo de bom!

Aqui, reunimos alguns dos motivos mais comuns, imaginando que você possa querer adicioná-los à sua lista.

### 2.1 DINHEIRO (Mercado de trabalho)

Ir bem em competições de programação indica que você programa muito bem, uma habilidade muito desejada pelas empresas de tecnologia. Empresas patrocinam e até fazem os seus próprios eventos de programação para buscar talentos. Maratona é emprego garantido, e as grandes empresas de tecnologia estão cheias de ex-maratonistas.

### 2.2 Viagens

As competições te permitem conhecer muitos lugares diferentes. Como cada etapa é em um lugar, quanto melhor seu desempenho mais longe você poderá ir! Veja por onde a UFPR já passou:

- Florianópolis
- Belo Horizonte
- Vila Velha
- Estocolmo, Suécia
- Campinas
- Joinville
- Orlando, EUA
- Goiânia
- Varsóvia, Polônia
- Londrina
- Ponta Grossa
- Uberlândia

- Fortaleza
- Foz do Iguaçu
- Campo Grande

## 2.3 Camisetas e balões

Veja estes competidores da UFPR. Veja como estão felizes.



## 2.4 Resolução de problemas

Na programação competitiva, você é levado a trabalhar com muitos problemas, todos muito diferentes. Pensar no melhor algoritmo para resolvê-los lhe traz melhoras no foco e na resolução de problemas.

## 2.5 Conceitos da computação

Estudando para a maratona, você encontrará bem cedo conceitos que o curso de graduação só vai oferecer mais tarde; isso implica em facilidade em algumas disciplinas que não deviam ser tão fáceis assim, isso sem contar em aprender coisas que os professores não vão nem mencionar!

## 2.6 Trabalho em equipe

A maratona da SBC e da ICPC é obrigatoriamente feita em grupos de três pessoas, mas com recursos escassos. Cada equipe só tem acesso a um computador e, por isso, vocês precisam se organizar para usar o tempo de melhor maneira no desenvolvimento, digitação, teste e envio de suas soluções.

Restrições assim permitem que uma boa dinâmica de equipe seja construída pelos colegas, além do fortalecimento de habilidades de trabalho em grupo que são úteis fora da maratona.

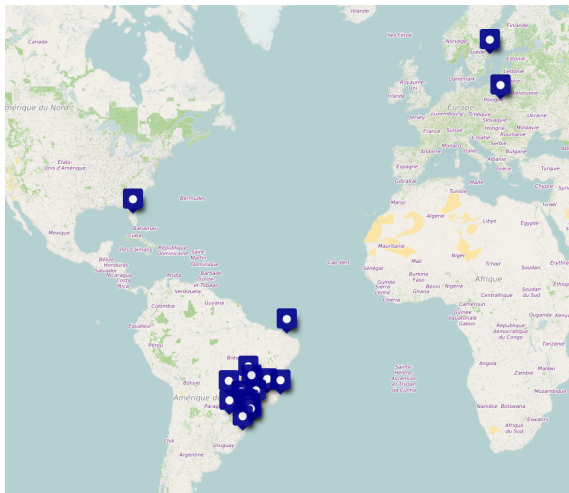
## 3 UFPR na Maratona

A UFPR tem história na maratona. Nós já fomos para 14 finais nacionais e 3 finais mundiais.

Se o pessoal que estudou aqui 10 anos atrás - comendo no mesmo RU, tendo aula com os mesmos professores, usando os mesmos laboratórios - conseguiu, você também consegue!







## 4 Como treinar

### 4.1 Preciso saber alguma linguagem específica?

As linguagens de programação que são usadas em competições são muito variadas. Na Maratona da SBC, que é nosso foco, são aceitas cinco: **C++**, **Java**, **Python**, **C** e **Kotlin**. E apesar de muitas pessoas utilizarem Python para problemas simples que não precisam de muito desempenho, as linguagens mais populares são C++ e Java.

Aqui na UFPR (e, na verdade, em boa parte do Brasil) a preferência é por C++. Isso porque ela vem com muitas estruturas úteis na sua biblioteca padrão, livrando você de digitar muito código. E note que no curso de BCC **não será ensinado C++**, apenas C, que é a base – mas você poderá fazer boa parte dos trabalhos em C++ e terminar antes dos seus colegas.

**E se você não sabe programar, não se preocupe!** Todo mundo esteve em sua posição um dia. Nas próximas seções, você encontrará dicas para aprender a usar C++ para programação competitiva.

## 4.2 O que eu posso levar e usar na competição?

Apenas material impresso, assim como livros e apostilas. O que levar faz parte da estratégia da sua equipe, pois não há acesso à internet, e as ferramentas disponíveis são as mesmas para todos os competidores.

Sempre, o ambiente trará um editores de texto no terminal como `vim` e o `emacs`, além do clássico `gedit`. Recentemente, as competições também contam com IDEs da JetBrains (ex. `CLion`, `PyCharm`), e editores novos (ex. `VS Code`, `Sublime`).

## 4.3 Formato de uma competição genérica

Todas as competições de programação seguem um padrão claro e simples. Sempre consistem em um número de problemas, não necessariamente em ordem de dificuldade, os quais você pode responder em qualquer ordem.

Os problemas são de entrada e saída textuais, isto é, em todos eles você deve escrever um programa que lê uma entrada em formato específico, faz alguma coisa e escreve uma saída em outro formato também especificado.

Nenhum dos desafios exige que você crie janelas bonitas, uma página web ou que você mexa com impressoras. O foco é nos algoritmos e nas estruturas de dados, sendo que não é permitido fazer uso de programas do sistema (por exemplo, chamar o `grep`).

A resposta que você submete pode ser uma de duas (e aí depende da competição): o código-fonte — o programa — ou

a resposta para um problema ou enigma. E você verá que o primeiro caso é bem mais comum.

No caso da entrega do código-fonte, o seu programa é executado e avaliado por um juiz automatizado (e existem muitos desses juízes online). Ele testará seu programa com diversas entradas e então lhe entregará um resultado, que será um dos seguintes:

### **AC** – Accepted

Seu programa passa em todos os casos de teste;

### **WA** – Wrong Answer

Seu programa entregou uma resposta incorreta em pelo menos um dos casos de teste;

### **TLE** – Time Limit Exceeded

Seu programa demorou para responder um caso de teste dentro do limite de tempo que o problema aceita;

### **CE** – Compilation Error

Seu programa deu erro na hora da compilação, lembrando que *warnings* não são erros;

### **RE** – Runtime Error

Seu programa falhou durante a execução. Isso significa um acesso de memória inválida, uma divisão por zero ou qualquer saída não-zero da função principal.

E isso é tudo. Na competição, o juiz não vai te mostrar o teste que deu errado, ou qual linha do seu programa deu erro: descobrir o porquê é sua responsabilidade! Mas não se preocupe, você poderá arrumar seu código e submeter ele de novo.

Seu objetivo é alcançar o máximo de respostas corretas ou ACs e evitar os erros, porque eles te trazem penalidades que são usadas para desempatar competidores.

## 4.4 Online Judges

**Online Judges** (OJs) são sistemas juiz que te dão problemas, recebem submissões e testam o seu programa, sendo que eles também são usados nas competições.

Para um aprendizado mais uniforme da maratona, recomendamos que você utilize vários simultaneamente. Cada juiz tem um conjunto de problemas diferente com fórmulas e formatos diferentes. Ficar preso a apenas um juiz vai limitar a extensão dos problemas que você consegue resolver.

### 4.4.1 Codeforces

O **Codeforces** (CF) é um site russo muito famoso entre os praticantes da programação competitiva. Contando com milhares de problemas, nele acontecem toda semana diversos *contests*, que são competições virtuais.

O foco dele é em problemas lógicos e de raciocínio matemático. Principalmente nos problemas mais fáceis, o uso de estruturas de dados mirabolantes vai ser escasso. Ele é ótimo para desenvolver um raciocínio matemático rápido, mas não é exatamente o ideal para se aprofundar em diversos assuntos.

Cada usuário tem um **rating**, que é alterada (para mais ou para menos) depois da participação em *contests*. Essa alteração funciona num sistema elo e depende tanto do desempenho do usuário quanto de todos os outros participantes do *contest*.

E depois dos *contests*, o Codeforces sempre libera um **editorial**: uma explicação de solução para cada um dos problemas (muitas vezes com exemplos de implementações), que também lhe ajuda a descobrir novas técnicas.

O Codeforces também conta com uma funcionalidade de **gym**, que permite que qualquer pessoa coloque problemas na plataforma. Nesses conjuntos de problemas que são parale-

los aos do site, podemos encontrar todo tipo de competição, incluindo os da SBC/ICPC.

Além disso, o site conta com uma seção educacional novíssima com um curso muito bom. O curso da [Universidade ITMO](#) ensina algumas estruturas de dados e algoritmos extremamente úteis para qualquer competidor.

#### 4.4.2 CSES

O **CSES** é um juiz finlandês que acompanha os conceitos do livro do Laaksonen, [Competitive Programmer's Handbook](#) (gratuito). Os problemas são de conceito único, sendo bem úteis pra aprender diversas técnicas diferentes.

Porém, diferente do juiz UVA (que você conhecerá a seguir), o livro não contém referências ao CSES, porém todos os problemas encontrados no livro estão no repositório de problemas do CSES.

#### 4.4.3 AtCoder

O **AtCoder** é um juiz japonês muito semelhante ao CF, com ainda mais foco em problemas matemáticos.

E além disso, o AtCoder sedia semanalmente os **AtCoder Beginner Contests**, ou **ABCs**, que têm uma curva de dificuldade mais adequada para iniciantes do que a do Codeforces.

#### 4.4.4 BeeCrowd

O **BeeCrowd** é um juiz brasileiro que oferece todos os problemas das competições anteriores da SBC/ICPC, OBI e outras competições, assim como um catálogo diversificado de problemas, sendo todos eles formatados de forma mais parecida com o das competições da maratona.

O grande problema é que ele não é modelado para facilitar o aprendizado, sem editoriais, soluções e com poucas discussões. Porém, ele é ótimo principalmente para praticar a leitura dos problemas da SBC que trazem uma historinha de alguns parágrafos que serve apenas para te distrair, um formato diferente daquele encontrado no Codeforces.

Ele tem uma organização em categorias, incluindo uma categoria de **Iniciante** que pode ser útil para focar os seus estudos.

#### 4.4.5 UVA

O **UVA** é um juiz que acompanha o livro dos irmãos Halim, **Competitive Programming**. Os seus problemas são no formato da maratona tradicional e ensinam bastante, porém é um repositório impossível de navegar sem o livro.

O livro conta com dicas de como resolver os problemas, a solução de muitos e uma lista categorizada de problemas para resolver para treinar cada conceito ensinado.

## 4.5 Como aprender

Cada pessoa aprende de um jeito, mas existem alguns padrões:

### Praticar

O único jeito de fixar o seu conhecimento é praticando. O conceito mais difícil não é nada perto dos problemas onde ele é aplicado: por isso, pratique. É normal ter receio de errar e falhar, mas a maratona te ensina que isso faz parte do aprendizado. Seu objetivo é melhorar, então vá lá e se esforce para isso! Você pode:

### Fazer contests

Se colocar num ambiente parecido com o de competição com um tempo limite e com pessoas ao seu

redor te prepara para resolver problemas rapidamente.

### **Organizar e fazer uma lista de problemas**

Vários competidores tem uma planilha com diversos problemas que eles desejam resolver porque foram recomendados por outros competidores. Geralmente os problemas têm certas peculiaridades ou várias maneiras de serem resolvidos, sendo um bom exercício para praticar.

### **Dar aula, escrever**

O melhor jeito de ter certeza que aprendeu é ensinando seus colegas. Na hora de formular as explicações, o conteúdo fixará melhor na sua cabeça e as dúvidas de seus colegas levará você a pesquisar coisas diferentes. Não tenha medo de não saber alguma coisa, ninguém nasceu sabendo de tudo!

### **Estudar**

Saber exatamente como funcionam as diversas estruturas de dados, a teoria dos números, combinações e grafos te ajudará muito em como resolver problemas. Todo ano, por exemplo, a Unicamp organiza uma escola de verão para ensinar diversas técnicas diferentes, e as quais as aulas você pode encontrar no YouTube na próxima seção. Em geral, pode se dizer que a maratona é só uma desculpa pra estudar.

### **Upsolve**

Se você não conseguir resolver um problema, espere um pouco antes de buscar a solução! Tente abstrair o problema por alguns minutos ou horas para descobrir exatamente qual é a essência do problema que você precisa

resolver. Depois de pensar bastante, a solução que outra pessoa escreveu vai encaixar como uma luva, porque aí você poderá equipar o seu arsenal com uma técnica nova, que você poderá testar pra ver se resolve um problema menos específico do que o original.

## Refletir

Esqueça o problema por um tempo, dê um tempo para o seu cérebro descansar. Depois de alguns dias ou até mesmo no mesmo dia, uma ideia nova pode aparecer para você no banho, no almoço ou no caminho de casa. Você se surpreenderá!

## 4.6 Onde aprender

### 4.6.1 Livros

#### **Laaksonen, Antti. Competitive Programmer's Handbook**

Livro excelente, um pouco incompleto, porém atualizado e gratuito. Os problemas do livro estão no CSES.

#### **Felix Halim, Steven Halim. Competitive Programming 4**

Livro excelente, com muitas dicas e com exercícios no UVA.

### 4.6.2 Canais Educacionais no YouTube

#### **MaratonUSP (em português)**

Aulas do grupo de maratona da USP-IME com aulas para iniciantes. A *playlist* “Bixecamp” por exemplo ensina o básico de C++ e programação.

#### **GEMA ICMC (em português)**

Aulas do grupo de maratona do ICMC-USP.



## Clube de Programação da UTFPR-CT (em português)

Aulas da oficina de programação da UTFPR, com uma excelente *playlist* de jornada de iniciação para iniciantes e vídeos sobre a resolução de diversos problemas.

## Maratona UFMG

Aulas do grupo de maratona da UFMG.

## Algorithms Live!

Canal com várias aulas sobre o uso de diversos algoritmos e estruturas de dados.

## Second Thread

Canal de um maratonista.

## Errichto

Canal de um grande competidor.

## Aulas de algoritmos da Universidade de Harvard

Conteúdo de computação com explicação de algoritmos e estruturas de dados avançadas.

## Aulas do curso de verão da Unicamp

Todas as aulas de preparação para as finais da maratona dadas na Unicamp.

## Abdul Bari

Aulas de algoritmos muito boas, muito bom para aprender KMP.

## Gaurav Sen

Resoluções de problemas específicos de maratona.

## Felipe Mota (em português)

Canal de um competidor brasileiro com resolução de problemas do BeeCrowd.

### **Giovanna Kobus Conrado (em português)**

Canal de uma competidora brasileira que explica conceitos diversos com foco na maratona da SBC.

### **Naum Azeredo (em português)**

Canal de um competidor brasileiro com conteúdo em alguns tópicos avançados de maratona.

### **Bruno Monteiro (em português)**

Canal de um competidor brasileiro com conteúdo em alguns tópicos avançados de maratona.

## **4.6.3 Outros recursos**

### **CP-Algorithms e CP Algoritmos em Português**

Enciclopédia de algoritmos excelente com diversos tópicos em grande detalhe.

### **USACO Guide**

Guia completo para a olimpíada de informática americana.

### **Codemonk**

Curso de fundamentos de programação em C++.

### **Neps Academy (em português)**

Cursos (incluindo C++ para programação competitiva), listas de exercício e discussão. Definitivamente o site mais bonito que você vai ver listado neste livreto.

### **Curso do NOIC (em português)**

Curso de informática para olimpíada do Núcleo Olímpico de Incentivo ao Conhecimento.

### **Geeks for Geeks**

Um pouco de tudo, rico em conteúdo; Inclui tutoriais

para aprender C++. Cuidado com implementações erradas!<sup>1</sup>

## C++ Reference

Site de referência para as bibliotecas padrão do C++.

## uDebug

Site com entradas e saídas submetidas por usuários para testar os seus programas com suporte a vários judges diferentes.

## Stanford: Introduction to Programming Contests

Aula de Stanford de introdução a maratona com muitos slides e conteúdo bom.

## TopCoder: Competitive Programming Tutorials

Uma porção de tutoriais de maratona.

## $|V| + |E|$ · Gráfico (em português)

Um repositório de dicas de maratona mantido pela equipe de maratona da UFPR.

## Ideias do NOIC (em português)

Lista de ideias de maratona mantido pelo NOIC.

## UCCPS (em inglês)

Canal da sociedade de programação competitiva da Universidade de Cambridge com palestras de pessoas importantes (Steven Halim, Antti Laaksonen, etc).

---

<sup>1</sup>Exemplos de implementações erradas: Ponto em polígono, Aho-Corasick, além de análises assintóticas incorretas. Tome cuidado!

## 4.6.4 Outros OJs

### ICPC Live Archive

Arquivo com problemas usados nas regionais e nas finais da ICPC.

### SPOJ Brasil e SPOJ

Juiz antigo com uma infinidade de problemas diferentes.

### CodeChef

Juiz com contest mensais com muitos problemas excelentes

### VJudge

Meta-OJ que agrupa os problemas de vários outros OJs em uma só plataforma para realizar competições.

### TopCoder

OJ de competição para empresas.

### Timus

Arquivo russo com muitos problemas.

### Meta Hacker Cup

Competição de programação da Meta.

### LeetCode

Uma plataforma de aprendizado para se preparar para entrevistas técnicas.

### HackerRank

Outra plataforma para entrevistas técnicas.