

 01

Introdução

Transcrição

Bem-vindos ao curso de Maratona de Programação! Neste curso focaremos em três problemas reais de sites distintos de maratonas de programação, e os resolveremos usando o Java. Portanto, a linguagem Java é um pré-requisito, mas é o único. Se você programa em outras linguagens e gostaria de ver esse curso mesmo assim, tudo bem. Se você sentir que seria melhor aprender Java antes, vá para [os nossos cursos de Java](https://cursos.alura.com.br/career/desenvolvedor-java-junior) (<https://cursos.alura.com.br/career/desenvolvedor-java-junior>) e volte aqui depois.

A intenção não é apenas resolver os problemas de maratona correndo, com um código meia-boca. Nas competições, sejam elas regionais, nacionais ou mundiais, o desafio não é escrever um código que resolva o problema. Quando trabalhamos em uma equipe de programadores dentro de uma maratona mundial de programação, estamos interessados em resolver o problema de um cliente. Temos um problema a ser resolvido, um algoritmo simples, complexo ou muito complexo, com suas respectivas peculiaridades. O cliente apresenta esse problema, explicitando algumas coisas, mas há também detalhes implícitos que não conseguimos perceber tão facilmente. Por vezes há também algumas ambiguidades com as quais nós, programadores, precisamos saber lidar no cotidiano.

Enfrentamos todas essas questões em uma maratona de programação. Durante os 7 anos em que competi, sempre que precisava enviar uma submissão para mostrar que o meu programa estava resolvendo o problema, eu não podia correr o risco de o cliente dizer que o programa não funcionava em determinado caso. Isso tiraria pontos e eu perderia a chance de ir para o mundial representar o Brasil. Assim, quando vamos programar em uma maratona, e queremos realmente ir para campeonatos maiores, temos que pensar na qualidade do código. E essa qualidade deve ser entregue em diversos sentidos: o código precisa resolver todos os problemas do cliente, tudo o que ele pediu explicitamente e o que está implícito, o que mais vier junto com o problema, precisamos antecipar e resolver.

É igual ao nosso dia a dia de desenvolvedor de software: recebemos pedidos dos nossos clientes, com detalhes explícitos e implícitos, e temos que saber nos questionar se o algoritmo que estamos escrevendo, seja ele complexo ou simples, tem algum tipo de falha. E nas maratonas de programação, a margem de erro é 0. Se errarmos, perdemos pontos na lata. Assim como, na vida real, se entregarmos um projeto que não funciona para um cliente, ele reclamará com você.

Então a maratona e os exercícios de maratona devem te fornecer o *mindset* de "Eu preciso resolver, e preciso resolver bem de primeira", que é extremamente importante para o cotidiano do desenvolvedor. E é isso que eu quero passar aqui, até mais que os algoritmos em si. Muito mais do que aprender algoritmos interessantes, nesse curso você vai aprender a pensar com a cabeça de um programador sênior. A pensar como proteger o código para que o cliente tenha sua demanda atendida, usando exemplos de maratona de programação. São exemplos que envolvem algoritmos e ajudam a desenvolver a nossa cabeça e os nossos dedos na hora de desenvolver o código. E note que não falamos em qualidade de código no mínimo de caracteres possível. Em boa parte do mundo, a quantidade de código conta, o número de letras conta. E, de propósito, vamos trabalhar um exemplo em que isso faz diferença. No mundo real, você não está sendo pago para digitar menos. Você está sendo pago para resolver muito bem um problema. Na maratona de programação, a lógica é a mesma; você não está lá para escrever as variáveis `a`, `b`, `c`, `d` e `e`. Será verificado se a sua solução resolve mesmo o problema do cliente em todos os casos possíveis, com a qualidade suficiente. É dessa maneira que somos testados em uma maratona, e é dessa maneira que somos testados no mundo real.

Eu vou mostrar três problemas, e vamos resolvê-los com esse *mindset* devagar, vendo o passo a passo. Depois vou resolvê-los como eu fazia durante as maratonas, para mostra a você que realmente quer competir, como era quando eu participava de uma equipe. Hoje em dia, que já não posso participar por causa do limite de idade, eu trabalho com a mesma cabeça, querendo enviar o mínimo de bugs possível para o meu cliente. Então, o *mindset* aqui é o mais importante - e de quebra aprenderemos alguns algoritmos e a entender melhor o cliente. Depois desse curso, há muitos

outros pelos quais você pode passar na Alura, por exemplo na parte de [algoritmos](#) (<https://cursos.alura.com.br/course/projetos-de-algoritmos-1>), e para quem já viu alguns algoritmos, os de *machine learning*, para quem quiser ver temos também de [estrutura de dados](#) (<https://cursos.alura.com.br/course/estrutura-de-dados>) e de [estatística](#) (<https://cursos.alura.com.br/course/introducao-a-estatistica-1>). São as opções mais interessantes para depois que você completar esse curso. Então vamos começá-lo! Bons estudos!