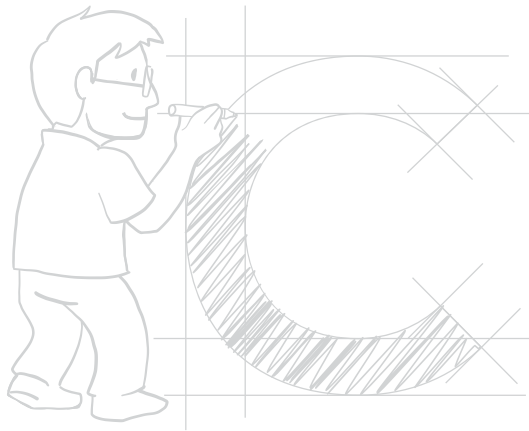


INTRODUÇÃO À PROGRAMAÇÃO COM A LINGUAGEM C

Rodrigo de Barros Paes



© Novatec Editora Ltda. 2016.

Todos os direitos reservados e protegidos pela Lei 9610 de 19/02/1998. É proibida a reprodução desta obra, mesmo parcial, por qualquer processo, sem prévia autorização, por escrito, do autor e da Editora.

Editor: Rubens Prates

Assistente editorial: Priscila A. Yoshimatsu

Revisão gramatical: Solange Martins

Capa: Carolina Kuwabata

Editoração eletrônica: Carolina Kuwabata

ISBN: 978-85-7522-485-4

Histórico de impressões:

Março/2016 Primeira edição

Novatec Editora Ltda.

Rua Luís Antônio dos Santos 110

02460-000 – São Paulo, SP – Brasil

Tel.: +55 11 2959-6529

Email: novatec@novatec.com.br

Site: www.novatec.com.br

Twitter: twitter.com/novateceditora

Facebook: facebook.com/novatec

LinkedIn: linkedin.com/in/novatec

CAPÍTULO 1

Uma palavra do autor

1.1 Caro estudante

Olhe ao seu redor. Smartphones, relógios inteligentes, carros que se conectam à Internet, bancos que operam praticamente via Internet, TV Digital, serviços de vídeo on-line, aplicativos que controlam a rota de veículos, aplicativos para táxis, enfim, o mundo hoje é digital.

Todas essas inovações só se tornaram realidade graças à programação de computadores. A programação é a habilidade que permite controlar como cada um desses serviços irá funcionar. De certa forma, dominar essa habilidade é ganhar poderes de fazer coisas incríveis em um mundo digital.

Obviamente, você não se tornará um gênio da programação do dia para noite. Será preciso muito estudo e dedicação. Mas tenha certeza de que o caminho poderá ser bastante prazeroso. Cada descoberta, cada habilidade adquirida aumentará o seu poder de fazer mais e mais nesse mundo digital.

Este livro é o seu primeiro passo nessa grande caminhada. Ele aborda os princípios da programação em detalhes. Faremos isso não pelo caminho mais fácil, mas pelo caminho mais duradouro. Depois de concluir este livro, você estará apto a partir para novos desafios usando as tecnologias que existem atualmente e também estará preparado para as que virão.

Para isso, estruturamos o livro em uma abordagem baseada na resolução de muitos problemas utilizando a linguagem C. Os problemas lhe darão uma capacidade de resolver novos problemas complexos quando eles se apresentarem para você. Já a linguagem C lhe dará o conhecimento necessário para entender como os programas funcionam “por debaixo dos panos”. Com isso, se você quiser utilizar outra

linguagem depois de concluir este livro, já saberá que não existe mágica e que todos os benefícios dessa nova linguagem têm um custo que você saberá avaliar.

Reserve tempo para praticar. Procure se divertir durante o processo. E seja muito bem-vindo ao mundo da programação de computadores!

1.2 Caro professor

O que leva alguns estudantes a se saírem tão bem nas disciplinas de programação, enquanto outros sofrem para conseguir entender o mínimo? Intuitivamente, sabemos que a prática de exercícios de programação leva ao aprendizado. Vários artigos científicos corroboram essa ideia¹². Sou professor de programação e em 2009 resolvi fazer com que os meus alunos praticassem mais. Elaborei e cobrei a resolução de uma série de problemas, aproximadamente vinte por semestre. Porém, na época, a minha turma de alunos era composta de quarenta alunos e eu tinha duas turmas, isso me deixava com 1.600 códigos-fonte para corrigir durante um semestre letivo. Muitas vezes eu demorava duas semanas para dar uma resposta ao aluno sobre a tentativa dele. Esse é um prazo muito longo para um período letivo tão curto e muitas vezes causa um desestímulo e mesmo uma dificuldade de compreender os assuntos seguintes. Pensando nisso, idealizei um juiz on-line focado no auxílio às aulas de programação. Esse juiz está disponível em www.thehuxley.com.

Os benefícios do uso da ferramenta foram rapidamente percebidos, em especial com o aumento substancial da prática. Entretanto, da mesma forma que antes, ainda existiam alguns alunos que, mesmo com a ferramenta, continuavam com um desempenho abaixo do esperado. Então, intensifiquei a pesquisa sobre o problema e comecei a refletir sobre o material didático e a forma como as aulas eram lecionadas. Foi quando deparei com vários trabalhos que abordavam dois problemas: (i) Como os alunos aprendem programação? (ii) Alunos diferentes aprendem de forma diferente?

Este livro é o resultado da aplicação do conhecimento obtido na tentativa de responder a essas duas perguntas. Duas ideias principais norteiam a metodologia aplicada, as quais estão descritas a seguir.

1 Robins, A., Rountree, J., Rountree, N. Learning and teaching programming: a review and discussion. *Computer Science Education*, Vol. 13, Iss.2, (2003).

2 Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Devlin, M., Paterson, J., A survey of literature on the teaching of introductory programming. *SIGCSE Bull.* 39, 4 (2007).

- **Existem diferentes tipos de aprendiz.** Algumas pessoas aprendem melhor quando leem, outras quando são apresentadas à imagens e algumas quando colocam a “mão na massa”. Procuramos apresentar os conceitos de forma que esses diferentes perfis possam se sentir confortáveis. O livro conta com a explicação dos conceitos, figuras e representações que os ilustram e muitos exemplos de aplicações práticas, além de uma série de exercícios propostos.
- **O conhecimento é construído por meio do desenvolvimento de várias habilidades.** Por exemplo, existe a habilidade de encontrar um elemento em uma lista de valores, ou a habilidade de ordenar uma lista, a habilidade de escrever algo na tela, e assim por diante. Com o tempo, essas habilidades se tornam cada vez mais abstratas, por exemplo, a habilidade de resolver um problema usando “dividir e conquistar”. A diferença entre um programador iniciante e um experiente se dá principalmente em quais conjuntos de habilidades cada um possui. O desenvolvimento de uma habilidade se baseia no aprendizado a partir da experiência. Esse aprendizado se dá por meio de quatro etapas, ilustradas também na Figura 1.1:

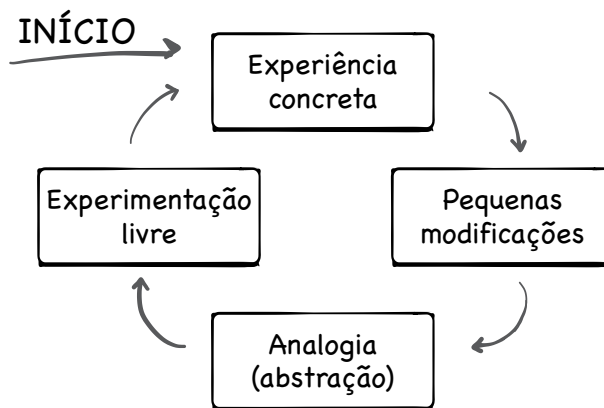


Figura 1.1 – Ciclo de aprendizado. Fonte: Adaptado de: Kolb, D. *Experiential Learning: experience as the source of learning and development*. Prentice Hall, 1985.

1. **Experiência concreta.** Nessa etapa, mostramos um exemplo concreto ou um problema no qual a habilidade foi aplicada.
2. **Observação e reflexão por meio de pequenas modificações.** Nesse passo, o aprendiz, com base na experiência concreta anterior, deve modificá-la e refletir sobre cada alteração. Nesse caso, propomos uma série de exercícios que são pequenas alterações em resoluções já apresentadas no livro. É a partir dessa etapa que o aprendiz começa a praticar.

3. **Abstração e aplicação em outros contextos.** O aprendiz deve levantar hipóteses e tentar aplicar a habilidade em contextos diferente dos mostrados anteriormente. Com isso, ele percebe que aquela habilidade pode ser útil para resolver problemas diferentes dos que vivenciou na experiência concreta. Nesse caso, preparamos exercícios com o mesmo nível de dificuldade dos exercícios resolvidos no livro. Assim, o aluno pode por analogia resolvê-lo e se sentir mais confortável para passar para a próxima etapa.
4. **Experimentação ativa e livre.** Nessa etapa, o aprendiz já deve se sentir confortável com a nova habilidade de modo a enxergar a sua aplicação na resolução de problemas de forma livre. Para isso, separamos várias questões para cada tópico de forma que o aluno pode resolvê-las e submeter sua solução para um juiz on-line, que indicará se ele acertou ou errou.

Todos as habilidades deste livro foram estruturadas pensando em exercitar essas quatro etapas e, portanto, é importante que o aluno não “queime” etapas para que o conhecimento seja realmente profundo.

Assim, este livro é baseado em uma metodologia de aprendizado por experiência e com grande foco na prática de exercícios. São apresentados muitos exemplos, e os conceitos são introduzidos sempre que necessários para resolver um determinado problema, e não ao contrário.

1.3 Como utilizar este livro

Este livro foi idealizado para ser utilizado durante um semestre letivo como livro didático em disciplinas introdutórias de programação, seja em cursos técnicos, universidades ou mesmo em cursos de curta duração. Normalmente a ementa dessas disciplinas envolve todos os capítulos deste livro com exceção dos capítulos 15, 16 e 17.

O Capítulo 15 contém um catálogo com as principais habilidades de um programador iniciante. Esse capítulo serve como um guia de referência rápida para aquele momento em que você depara com um problema, já resolveu algo parecido antes, mas não lembra exatamente como foi.

O Capítulo 16 contém várias curiosidades e perguntas que os alunos geralmente fazem durante o decorrer do curso. Com frequência essas perguntas são mais relacionadas com detalhes da linguagem e não são cruciais para a resolução de problemas. Mas muitos professores incluem alguns tópicos abordados nesse capítulo em suas ementas.

Já no Capítulo 17 é proposto um guia de estilo de codificação com práticas mais comuns dentre os programadores em C. É importante que os alunos já aprendam a programar de forma organizada e, por isso, o guia resumido do Capítulo 17 pode ser útil.

Se você é um professor, tente incluir na sua metodologia de aula a realização de exercícios práticos pelos alunos, seja por meio do sistema The Huxley, seja por qualquer outro método da sua preferência. Mas procure fazer com que os alunos pratiquem bastante. Também é muito importante fazer com que esses exercícios não sejam optativos, eles precisam ser parte integrante do seu método de avaliação; caso contrário, é pouco provável que a maioria dos alunos da sua turma efetivamente os resolva.

Se você é um aluno, procure ler o livro na sequência em que os capítulos são apresentados. Para cada exemplo ou exercício comentado, procure reproduzi-lo você mesmo no computador. Se está lendo esse texto no computador, evite copiar e colar os exemplos. Digite você mesmo. Depois, mude um pouco o exemplo apresentado. Procure entender bem como funciona o conceito antes de passar para o próximo. Resolva todos os exercícios propostos no livro. Veja que não são tantos. Lembre-se de que a prática o tornará um bom programador.

1.3.1 Convenções adotadas neste livro

<trecho de código>

Significa que o trecho de código é obrigatório. Os símbolos < e > não fazem parte do trecho de código em si.

[*trecho de código*]

Significa que o trecho de código é opcional. Os símbolos [e] não fazem parte do trecho de código em si.

Itálico

Indica novos termos, URLs, endereços de email, nomes de arquivo e extensões de arquivo.

Monoespaçado

Usada para código de programa, bem como dentro de parágrafos, para referir-se a elementos do programa, como nomes de variáveis ou de funções, bancos de dados, tipos de dados, variáveis de ambiente, declarações e palavras-chave.

Monoespaçado itálico

Exibe textos que devem ser substituídos por valores fornecidos pelos usuários ou por valores determinados pelo contexto.

1.3.2 Códigos do livro

Os códigos utilizados neste livro estão, em sua maioria, disponíveis para download em <https://github.com/r0drigopaes/livro>. Eles foram nomeados de acordo com a legenda usada no livro. Por exemplo, o Código 4.1 está no GitHub como *4_1.c*.

1.4 The Huxley

Programação se aprende programando. E para programar mais e melhor é preciso ter desafios variados e em níveis de dificuldade crescente. De forma complementar aos exercícios deste livro, utilizaremos o The Huxley (www.thehuxley.com) para resolver problemas de programação. Os problemas do The Huxley vêm para preencher a etapa de “experimentação ativa e livre” adotada como metodologia para a escrita deste livro e, portanto, deve ser utilizado depois que o aluno já resolveu os exercícios propostos.

O The Huxley é uma ferramenta web que permite a alunos submeterem código em diversas linguagens de programação como respostas a exercícios de uma base de centenas de problemas. Para cada submissão, o aluno recebe feedback da correção automática pelo sistema por meio de uma série de testes e, em caso de erro, é direcionado para conteúdo de suporte.

Além dos alunos, os professores também são grandes beneficiados com a adoção desse tipo de ferramenta. Com o aumento do número de alunos em instituições de ensino, professores de Programação de Computadores e disciplinas afins são obrigados a investir grandes esforços na concepção de exercícios e, principalmente, na sua correção. Ainda que se utilizem monitores, trata-se de uma tarefa árdua e que, aos poucos, torna-se pouco personalizada de acordo com a real necessidade dos alunos – quanto mais alunos, menos personalizado o tratamento.

Com o The Huxley, o professor tem uma visão mais analítica e fidedigna do desempenho de seus alunos, incluindo quais estão com desempenho abaixo do esperado, a quantidade de problemas resolvidos, porcentagem de acertos/erros, tipos de problema com mais erros, detecção de plágio e erros específicos de cada

aluno. Em outras palavras, o professor passa a atacar de forma mais eficaz os problemas peculiares de aprendizado dos alunos, em vez de consumir seu tempo com a atividade de correção.

Por meio da interface do The Huxley, o professor cria questionários e avaliações a partir de uma base de centenas de exercícios existentes, podendo também criar seus próprios exercícios. Os exercícios incluem descrição e testes de aceitação do resultado. Os alunos acessam os exercícios definidos pelos professores, desenvolvem os programas e os submetem via interface do The Huxley. Para cada programa em linguagem específica, o The Huxley compila e testa o programa, armazenando o resultado para consulta posterior do aluno e do professor. O resultado é visualizado em forma de planilha e gráfico pelos professores, que podem atribuir nota diretamente com base na avaliação do sistema. Os alunos também recebem o resultado para que possam reenviar a solução caso haja algum problema de compilação ou nos testes de aceitação.

Do ponto de vista dos alunos, há ainda diversos mecanismos de incentivo que fazem com que eles tenham motivação para acessar o sistema diariamente, extrapolando a relação estudo-nota. Por exemplo, o Topcoder, que lista os alunos com maior quantidade de exercícios resolvidos, tem funcionado bem para garantir o sucesso do principal resultado do produto: a maior prática da programação.

O The Huxley é uma ferramenta gratuita para todos os alunos que querem praticar. Portanto, acesse, crie sua conta e prepare-se para aprender a resolver problemas com a programação de computadores.