

# 1. Bem vindo à Construção de Software

Você já conhece o significado de “construção” quando esse termo é usado fora do âmbito do desenvolvimento de software. “Construção” é a obra realizada por “operários da construção civil”, quando eles constroem uma casa, uma escola ou um arranha-céu. Quando criança, você provavelmente também deve ter construído coisas usando papel cartão ou algo assim. O termo “construção”, em seu uso predominante, identifica o processo de edificação. O processo de construção pode incluir alguns aspectos de planejamento, projeto e controle do trabalho realizado, mas “construção” refere-se principalmente à parte prática da criação de algo.

## 1.1. O que significa construção de software?

O desenvolvimento de software de computador pode ser considerado um processo complexo; nos últimos 25 anos, pesquisadores da área identificaram diversas atividades distintas relacionadas a essa prática. Entre elas, estão:

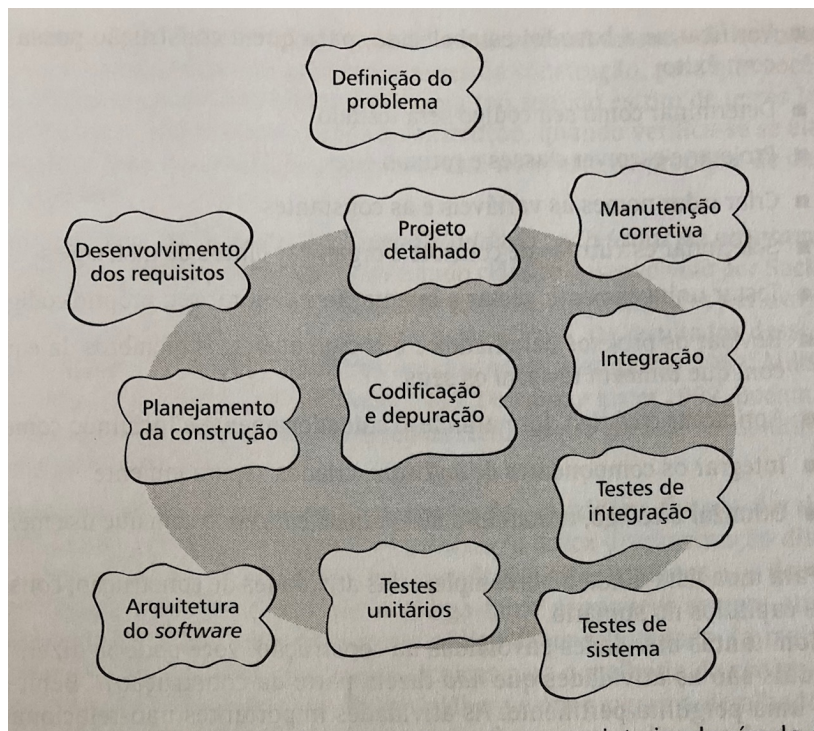
- ❖ Definição do problema
- ❖ Desenvolvimento dos requisitos
- ❖ Planejamento da construção
- ❖ Arquitetura do software ou projeto de alto nível
- ❖ Projeto detalhado
- ❖ Codificação e depuração
- ❖ Testes unitários
- ❖ Testes de integração
- ❖ Integração
- ❖ Testes de sistema
- ❖ Manutenção corretiva

Se você já trabalhou em projetos informais, talvez interprete que essa lista representa muita burocracia. Se já tiver trabalhado em projetos estritamente formais, então você terá certeza de que essa lista representa muita burocracia! É difícil estabelecer um equilíbrio entre pouca e muita formalidade.

Se você aprendeu a programar sozinho ou trabalhou mais frequentemente em projetos informais, talvez não tenha feito distinções dentre as muitas atividades relacionadas à criação de um produto de software. Mentalmente, você pode ter agrupado todas essas atividades como sendo a “programação”. Se você trabalha em projetos informais, a principal atividade em que pensa a respeito da criação de software provavelmente é aquela que os pesquisadores chamam de “construção”.

Essa noção intuitiva de “construção” é razoável, mas sofre de falta de perspectiva. Colocar a construção em seu contexto, junto com as demais atividades, ajuda a manter o foco nas tarefas corretas durante a construção propriamente dita e dá ênfase adequada a atividades importantes que não estão relacionadas a ela. A figura a seguir ilustra o lugar ocupado pela construção em relação às outras atividades de desenvolvimento de software.

Conforme indica a figura, a construção trata principalmente da codificação e da depuração, mas também envolve o projeto detalhado, o planejamento da construção, os testes da unidade, a integração, os testes de integração e outras atividades.



Aqui estão algumas das tarefas específicas envolvidas na construção:

- ❖ Verificar se a base foi estabelecida, para que a construção possa prosseguir com êxito
- ❖ Determinar como seu código será testado
- ❖ Projetar e escrever classes e rotinas
- ❖ Criar e dar nomes às variáveis e às constantes
- ❖ Selecionar estruturas de controle e organizar blocos de instruções
- ❖ Testar unitariamente. testar a integração e depurar seu próprio código
- ❖ Revisar os projetos detalhados e o código de outros membros da equipe e fazer com que também revisem os seus
- ❖ Aprimorar o código, formatando-o cuidadosamente e incluindo comentários
- ❖ Integrar os componentes de *software*, criados separadamente
- ❖ Otimizar o código, tornando-o mais rápido e fazendo com que use menos recursos

Com tantas atividades envolvidas na construção, você poderia dizer: “Tudo bem, mas quais são as atividades que não fazem parte da construção?” Bem, essa é realmente uma pergunta pertinente. As atividades importantes não-relacionadas à construção incluem o gerenciamento, o desenvolvimento de requisitos, a arquitetura do software, o projeto da interface com o usuário, os testes de sistema e a manutenção. Cada uma dessas atividades determina o sucesso final de um projeto, tanto quanto a construção - pelo menos, o sucesso de qualquer projeto que exija mais de uma ou duas pessoas e que dure mais do que algumas semanas.

## 1.2 Por que a construção de software é importante?

O simples fato de você estar lendo este livro já comprova que melhorar a qualidade do software e a produtividade do desenvolvedor é realmente muito importante. Grande parte dos projetos mais interessantes da atualidade utilizam software extensivamente. A Internet, efeitos especiais em filmes, sistemas médicos para

preservação da vida, programas espaciais, aeronáutica, análises financeiras de alta velocidade e pesquisa científica são alguns exemplos. Esses projetos e outros mais convencionais podem tirar proveito das práticas aprimoradas, pois muitos dos fundamentos são os mesmos.

Como você concorda que aprimorar o desenvolvimento de software é mesmo essencial, a pergunta que se apresenta é: por que a construção é tão importante? Aqui estão os motivos:

**A construção é uma parte extensa do desenvolvimento de software** - Dependendo do tamanho do projeto, a construção normalmente toma de 30 a 80% do tempo total. Qualquer atividade que tome tanto tempo por certo irá afetar o sucesso do projeto.

**A construção é a atividade central no desenvolvimento de software** - Os requisitos e a arquitetura são abordados antes da construção, para que você possa realizá-la eficientemente. Já os testes de sistema (no sentido estrito de testes independentes), obviamente, são realizados após a construção, quando verifica-se se ela foi executada corretamente. A construção, portanto, está no centro do processo de desenvolvimento de software.

**Com enfoque na construção, a produtividade Individual do programador pode aumentar significativamente** - Um estudo clássico mostrou que a produtividade individual de programadores variava segundo um fator de 10 a 20 durante a fase de construção.

**Frequentemente, o produto da construção - o código fonte - é a única descrição precisa do software** - Em muitos projetos, a única documentação disponível para os programadores é o código em si. As especificações de requisitos e os documentos do projeto podem ficar obsoletos, mas o código-fonte permanece sempre atualizado. Consequentemente, é imprescindível que o código-fonte seja dotado da mais alta qualidade possível.

**A construção é a única atividade que garantidamente será realizada** - O projeto de software ideal passa por um cuidadoso desenvolvimento de requisitos e projeto arquitetônico antes que a construção propriamente dita tenha início. O projeto ideal deve passar também por amplos testes de sistema, estatisticamente controlados, após a construção. Entretanto, projetos imperfeitos que encontramos no cotidiano frequentemente pulam a fase dos requisitos e do projeto do software, partindo diretamente para a construção. A fase de testes acaba sendo negligenciada, porque há erros demais para corrigir e o tempo torna-se escasso. Mas, independentemente de quão precipitado ou mal-planejado seja um projeto, você não pode omitir a construção; é onde "o pneu encontra o asfalto". Assim, aprimorar a construção é uma maneira de otimizar qualquer trabalho de desenvolvimento de software, não importa o quanto isso demore.

## Pontos-chave

- ❖ A construção é a atividade central no desenvolvimento de software; é a única atividade que garantidamente acontece em todo projeto.

- ❖ As principais atividades na construção são o projeto detalhado, a codificação, a depuração, a integração e os testes realizados pelo desenvolvedor (testes unitários e testes de integração).

- ❖ Outros termos encontrados para construção: "codificação" e "programação".

- ❖ A qualidade da construção afeta significativamente a qualidade do software.
- ❖ Em última análise, seu entendimento de como realizar a construção determina o quanto você é um bom programador, e esse é o assunto do restante deste livro.