

# O Programador Pragmático:

De Aprendiz a Mestre



Prefácio e Capítulo 1



## PET/ADS

**Este material foi desenvolvido pelo grupo  
PET/ADS do IFSP São Carlos**

# Prefácio

# O que te torna um programador pragmático?

**Ser Questionador:** Você tende a fazer perguntas. Isso é interessante – como o fez? Essa biblioteca lhe traz problemas? O que é esse BeOS do qual ouvi falar?

**Pensador crítico:** Raramente você aceita as coisas como lhe são passadas sem antes constatar os fatos.

**Adoção antecipada/adaptação rápida:** Você tem um faro para tecnologias e técnicas e gosta de testar coisas. Quando obtém algo novo, consegue entender rapidamente.

# O que te torna um programador pragmático?

**Realista:** Tente entender a natureza de cada problema que enfrenta. Esse realismo lhe dá uma boa percepção do nível de dificuldade das tarefas e quanto tempo elas levarão.

**Pau para toda obra:** Se esforce para conhecer um amplo conjunto de tecnologias e ambientes e tenta ficar a par de novos desenvolvimentos.



# Pragmáticos Individuais, Grandes Equipes

Na Engenharia de Software é muito importante a individualidade e a perícia profissional de cada membro da equipe envolvida no trabalho/projeto.

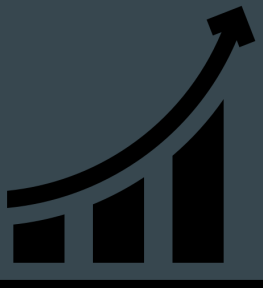
*“Daqui a cem anos, a engenharia atual parecerá tão arcaica quanto à usada pelos construtores de catedrais medievais são para os engenheiros civis de hoje, mas nossas habilidades pessoais ainda serão apreciadas.”*



# É Um Processo Contínuo

**Kaizen** é um termo japonês que captura o conceito de fazer continuamente diversas pequenas melhorias. Essa prática é atualmente usada em todo o mundo.

Trabalhe diariamente, aprimorando seus trabalhos e conhecimentos, para obter bons resultados futuros.



# Dicas



## DICA 1:

**Preocupe-se com seu trabalho.**

É inútil desenvolver software a menos que você se preocupe em fazê-lo bem.

## DICA 2:

**Reflita sobre seu trabalho.**

Nunca ligue o piloto automático. Reflita constantemente, criticando seu trabalho em tempo real.



# **Capítulo 1:**

## **Uma Filosofia Pragmática**

# O Gato Comeu meu Código-Fonte

Um programador pragmático toma as rédeas de sua carreira e não tem medo de admitir desconhecimento ou erro.

*“A maior de todas as fraquezas é o medo de parecer fraco”*

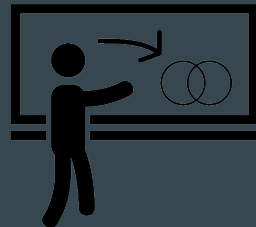
J. B. Bossuet



# Não use desculpas, forneça possíveis soluções

Sua desculpa esfarrapada nunca será convincente para ninguém. Por isso, planeje todos os planos de contingência possíveis caso algo dê errado.

Em vez de desculpas, forneça opções. Não diga que não pode ser feito, explique o que pode ser feito para salvar a situação.





## DICA 3:

### **Forneça opções, não dê desculpas esfarrapadas**

Antes de abordar alguém para dizer por que algo não pode ser feito, está atrasado ou ficou danificado, pare e escute a si próprio.

# Entropia de Software: Definição



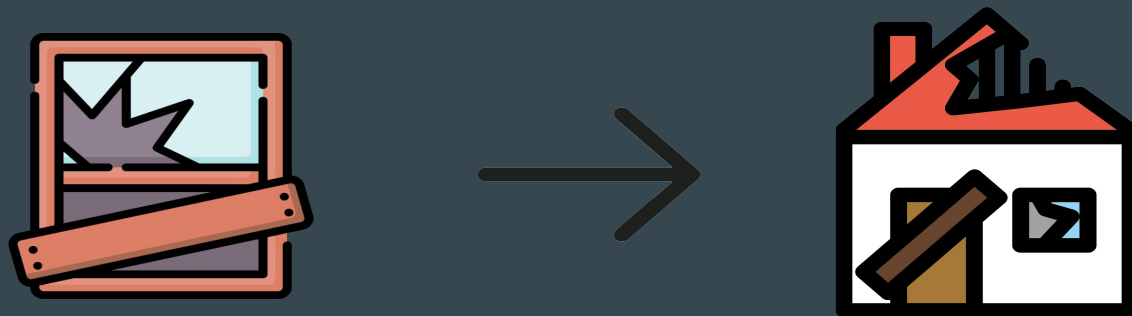
*“Entropia é um termo da física que se refere ao nível de  
“desordem” em um sistema”*

Quando a desordem aumenta no software, os  
programadores chamam isso de  
“deterioração de software”



# Teoria das janelas quebradas

Janela quebradas podem levar a deterioração de uma casa, o mesmo acontece nos códigos... uma parte esquecida do código acaba desestruturando-o por completo.



# Teoria das janelas quebradas

Por que isso ocorre?

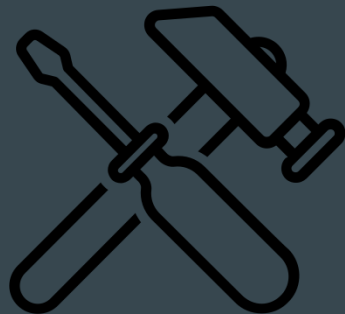
Uma janela quebrada cria a sensação de abandono nas pessoas, fazendo com que todos comecem a não se preocupar com o cuidado do local.



# Não contribua com a desordem!

Assim que ver uma janela quebrada, conserte-a para evitar problemas maiores.

Um pequeno trecho mal feito e esquecido do seu código pode acabar arruinando ele por completo.





# Apagando incêndios



O corpo de bombeiros foi acionado para apagar o incêndio que ocorria em uma casa muito bela e bem construída.

Lá os bombeiros se atentaram a não estragar o lindo carpete com suas mangueiras e então bolaram um plano para isso.

# Apagando incêndios



Ao se atentar em não sujar o carpete, os bombeiros apagaram o fogo sem causar novos problemas (por mais difícil que tenha sido).

Se esforce para consertar o problema, mas se atente aos resultados que isso pode causar, evitando novas complicações.



## DICA 4:

### **Não tolere janelas quebradas**

Não deixe “janelas quebradas” (projetos insatisfatórios, decisões erradas ou código inadequado) sem reparos.

# Sopa de Pedras e Sapos Cozidos

*“Os três soldados que voltavam da guerra estavam famintos. Quando viram a vila à frente, se animaram – tinham certeza de que os aldeões lhes dariam uma refeição. Mas, quando lá chegaram, encontraram as portas trancadas e as janelas fechadas. Após muitos anos de guerra, os aldeões estavam com pouca comida e estocavam o que tinham. Sem medo, os soldados ferveram água em uma panela e colocaram cuidadosamente três pedras nela. Os aldeões pasmos vieram observar. “Essa é uma sopa de pedras”, os soldados explicaram. “Isso é tudo que vocês colocam nela?” perguntaram os aldeões. “Exatamente – embora algumas pessoas digam que ela fica ainda melhor com algumas cenouras...” Um aldeão pôs-se a correr, voltando rapidamente com uma cesta de cenouras de seu estoque. Alguns minutos depois, os aldeões perguntaram novamente “É só isso?” “Bem,” disseram os soldados, “algumas batatas lhe dão sustância”. Outro aldeão saiu em disparada. Durante a hora que se seguiu, os soldados listaram mais ingredientes que melhorariam a sopa: carne, alho, sal e ervas. A cada vez um aldeão diferente corria para procurar em seus estoques pessoais. Eles acabaram obtendo uma grande panela de sopa fumegante. Os soldados removeram as pedras e se sentaram com a vila inteira para apreciar a primeira refeição reforçada que qualquer um deles tinha comido em meses”*



# Seja um catalisador de mudanças

Os soldados agem como um catalisador, reunindo a vila para que possam produzir conjuntamente algo que não poderiam ter feito sozinhos.

Com um resultado sinérgico. Todos acabam ganhando.

As pessoas acham mais fácil se associar a algo que vem obtendo êxito.

Mostre a elas um vislumbre do futuro e fará com que colaborem.



# Sapos Cozidos

Dizem que se você pegar um sapo e jogá-lo em água fervendo, ele saltará imediatamente para fora de novo.

Se você colocar o sapo em uma panela com água fria e aquecê-la gradualmente, o sapo não notará o lento aumento na temperatura e permanecerá quieto até ser cozido.



# Não seja um sapo

Não espere até ser tarde demais para se atentar aos erros, trabalhe diariamente para notar possíveis problemas e não acabar sucumbindo a eles.

Observe que o problema do sapo é diferente da questão das janelas quebradas.

Na Teoria da Janela Quebrada, se perde a vontade de combater a entropia porque se percebe que ninguém se preocupa. O sapo simplesmente não nota a mudança.





## DICA 5:

### **Seja um catalisador da mudança**

Busque instigar e despertar a curiosidade para a melhoria nos outros.

## DICA 6:

### **Lembre-se do cenário em larga escala**

A acumulação de pequenas coisas desintegra a motivação e as equipes, então pense antes de sair do controle.



# Software Satisfatório

Se esforce para criar um **bom** software.

Como todos os produtos e serviços, o software também deve satisfazer os usuários de maneira eficiente. Afinal ninguém gosta de usar serviços ruins e mal feitos.



# Envolva seus usuários na tomada de decisões

Quando criamos software para as pessoas, lembramos de pedir requisitos. Mas você pergunta a elas o nível de satisfação que desejam que seu software tenha?

Dê algo para seu usuário o mais cedo para que você possa receber um feedback do seu trabalho, assim fazendo melhorias.

# Saiba quando parar

Não estrague um programa que funciona perfeitamente o enfeitando e aprimorando excessivamente.

O código nunca chegará na perfeição, então, saiba quando parar.

Passe adiante e deixe que seu código fale um pouco por si próprio.





## DICA 7:

### **Torne a qualidade parte dos requisitos**

O escopo e a qualidade do sistema que você vai produzir devem ser especificados como parte dos requisitos desse sistema.

# Sua Carteira de Conhecimentos

Invista em seu conhecimento assim como um investidor financeiro:

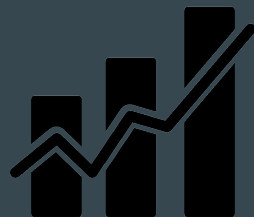
Invista regularmente

Diversifique

Gerencie o risco

Compre barato, venda caro

Reexamine e reestruture



# Construindo sua Carteira

## Invista Regularmente

Como nos investimentos financeiros, você deve investir em sua carteira de conhecimentos regularmente. Mesmo se for apenas um pequeno montante, o hábito é tão importante quanto as somas.

# Construindo sua Carteira

## Diversifique

Quanto mais coisas diferentes você souber, mais valor terá. A face da Computação muda rapidamente. Quanto mais tecnologias você conhecer, melhor conseguirá se adaptar à mudança.

# Construindo sua Carteira

## Gerencie o risco

A tecnologia se estende por um espectro de padrões que vai do arriscado e de alto retorno ao de baixo risco e baixo retorno.

Não arrisque todas as suas apostas técnicas na mesma direção, se arrisque em linguagens pouco usadas, mas, também, aprenda o que está na moda e funciona.



# Construindo sua Carteira

## Compre barato, venda caro

Aprender uma tecnologia emergente antes dela se tornar popular pode ser tão difícil quanto encontrar uma ação em baixa, mas os lucros podem recompensar.

Quem começou a aprender Java na década de 90 certamente tem uma boa bagagem e é bom nisso hoje.

# Construindo sua Carteira

## Reexamine e reestruture

Software é uma indústria muito dinâmica. Aquela tecnologia moderna que você começou a estudar no mês passado pode já estar ultrapassada.

Você pode ter de rever uma tecnologia que não usa há algum tempo. Ou, talvez, esteja mais apto àquela nova oferta de trabalho se aprender essa outra linguagem...

# Tenha Objetivos

**Aprenda pelo menos uma nova linguagem todo ano:** Aprendendo várias abordagens diferentes, você vai ajudar a expandir seu raciocínio e a evitar a estagnação.

**Leia um livro técnico a cada trimestre:** As livrarias estão cheias de livros técnicos sobre tópicos interessantes relacionados a seu projeto atual.

**Leia também livros não técnicos:** É importante lembrar que computadores são usados por pessoas. Não esqueça o lado humano da equação.

**Tenha aulas:** Procure cursos interessantes no colégio ou universidade de sua comunidade ou talvez na próxima feira que vier à cidade.

# Tenha Objetivos

**Participe de grupos de usuários locais:** Não vá e escute apenas, mas participe ativamente. O isolamento pode ser fatal para sua carreira.

**Experimente ambientes diferentes:** As livrarias estão cheias de livros técnicos sobre tópicos interessantes relacionados a seu projeto atual.

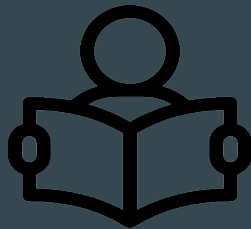
**Mantenha-se informado:** Assine revistas do ramo e outros periódicos. Selecione algumas que abordam tecnologias diferentes da de seu projeto atual.

**Mantenha-se conectado:** Grupos de notícias são ótimos para se descobrir que experiências as pessoas estão fazendo.

# Oportunidades de aprendizado

Se te perguntarem algo que você não tenha a mínima ideia, não deixe parar por aí, assuma como um desafio busque aprender.

Pergunte a um especialista, procure na web, vá à biblioteca. Se não conseguir achar a resposta sozinho, descubra quem pode encontrá-la. Não deixe sem resposta.



# Pensamento crítico

A última coisa importante é pensar criticamente sobre o que você lê e ouve.

Cuidado com os fanáticos que insistem que seu dogma fornece a única resposta. Ao menos se questione sobre outras possíveis soluções.

Nunca subestime o poder do comércio. Só porque um mecanismo de busca na Web lista um resultado antes não significa que essa seja a melhor solução.

# Dicas



## DICA 8:

**Invista regularmente em sua carteira de conhecimentos**

A dica mais importante para sua carteira de conhecimentos.

## DICA 9:

**Analise criticamente o que você lê e ouve**

Não apenas aceita as respostas e soluções dadas a você, mas analise criticamente.

# Comunique-se!



Não espere que em sua profissão não será necessário comunicar-se, pois terá que fazer muito isso.

*“Uma boa ideia fica órfã sem uma comunicação efetiva”*





# Como ter uma boa comunicação?

Saiba o que você quer dizer

Formular e dizer exatamente o que você pensou pode ser um grande desafio...

Então, faça um rascunho e pergunte a si próprio “*Isso representa o que estou tentando dizer?*”. Aperfeiçoe até que represente.

# Como ter uma boa comunicação?

## Conheça seu público-alvo

Você precisa conhecer as necessidades, interesses e capacidades de seu público-alvo.

Crie em sua mente um cenário fiel de seu público-alvo, o modo como você se comunica pode ser eficiente para um tipo de público e entediante para outro.

# Como ter uma boa comunicação?

## Escolha seu momento

Aborde as pessoas em momentos apropriados para que elas possam ser mais receptivas.

Torne o que você está dizendo relevante quanto ao momento, assim quanto ao conteúdo.

Às vezes, só precisamos da pergunta *“Esse é um bom momento para falarmos sobre...?”*

# Como ter uma boa comunicação?

## Escolha um estilo

Algumas pessoas preferem um resumo formal do tipo “apenas os fatos”.

Outras pessoas gostam de uma longa e variada conversa antes do assunto ser abordado.

Procure um melhor estilo de material. Adapte o estilo de sua fala para atender seu público-alvo.

# Como ter uma boa comunicação?

## Dê uma boa aparência

Suas ideias são importantes. Elas merecem um veículo de boa aparência para serem transmitidas para seu público-alvo.

Hoje em dia, não há desculpas para a produção de documentos impressos com aparência inadequada. Então, se dedique a apresentar algo “bonito” e bem feito.

# Como ter uma boa comunicação?

## Seja um ouvinte

Há uma técnica que você deve usar se quiser que as pessoas o escutem: escute-as.

Encoraje as pessoas a falar fazendo perguntas ou as faça repetir resumidamente o que disse a elas.

Transforme a reunião em um diálogo e obterá um resultado mais efetivo.

# Como ter uma boa comunicação?

## Dê retorno às pessoas

Quando fazemos uma pergunta a uma pessoa, achamos que ela é indelicada quando não responde.

Por mais que a resposta seja simples, manter as pessoas informadas faz com que elas não se importem tanto com o deslize ocasional e não se sintam esquecidas.

# O acróstico ASTUTO

O que você quer que eles **A**prendam?

Qual é **S**eu interesse no que você tem a dizer?

Seu **T**rabalho tem qual nível de sofisticação?

Que nível de detalhes eles **U**sam?

Quem você quer que **T**enha a posse da informação?

Como você pode motivá-los a **O**uvi-lo?



# Comunicação por e-mail

Tudo que foi discutido se aplica igualmente ao e-mail:

- Leia novamente antes de enviar;
- Verifique a ortografia;
- Mantenha o formato simples, sem estilizações desnecessárias;
- Tente usar o mínimo de repetição, não replique um email para dizer “concordo”;
- Ao usar citações feitas em emails de outras pessoas, certifique-se de dar os créditos;
- Não demonstre raiva;
- Verifique sua lista de destinatários antes de enviar;
- Arquive e organize seus emails.



## DICA 10:

**É o que você diz e a maneira como diz**

A menos que você trabalhe sozinho, tem de se comunicar. Quanto mais eficaz a comunicação, mais influente você será.

# Sobre os autores

Andrew Hunt trabalhou em diversas áreas, como telecomunicações, serviços financeiros, artes gráficas etc. Também se especializou em combinar técnicas já consolidadas com tecnologias de ponta, criando soluções novas e práticas. Ele administra sua empresa de consultoria em Raleigh, Carolina do Norte.

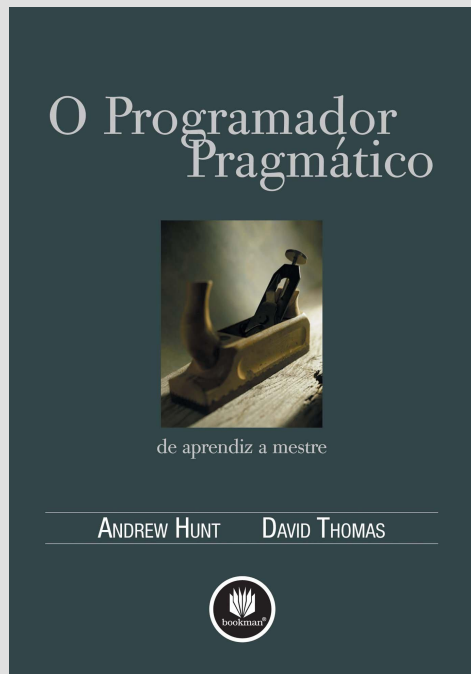
David Thomas, fundou em 1994, na Inglaterra, uma empresa de criação de software certificada pela ISO 9001, que distribuiu mundialmente projetos sofisticados e personalizados. Hoje, Thomas é consultor independente e vive em Dallas, Texas.

Atualmente, David e Andrew trabalham juntos em The Pragmatic Programmers, L.L.C

---

Este conjunto de slides foi elaborado a partir da obra:

HUNT, Andrew ; THOMAS, David. **O Programador Pragmático: de aprendiz a mestre.** Bookman, 2010.



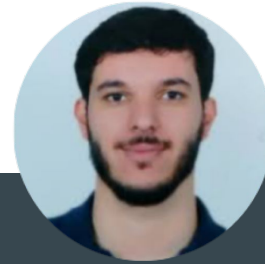
# A equipe



Felipe do Prado, Autor

---

Aluno e ex-bolsista do  
PET/ADS  
[LinkedIn](#)



Lucas Oliveira, Revisor

---

Professor de Computação, é tutor do  
PET/ADS desde janeiro de 2023.  
[LinkedIn](#)