



Os 10 PRINCÍPIOS

QUE TODO *BOM* PROGRAMADOR
DEVE SABER

POR MATHEUS BATTISTI
HORA DE CODAR

TABELA DE CONTEÚDOS

03	<i>INTRODUÇÃO</i>
05	<i>VOCÊ ESTÁ PRONTO?</i>
07	<i>A MELHOR LINGUAGEM</i>
09	<i>LINUX</i>
11	<i>FRAMEWORKS E LIBS</i>
13	<i>FERRAMENTAS PARA DEVS</i>
14	<i>SALÁRIOS VERSUS EXP</i>
15	<i>VAI FAZER FACULDADE?</i>
17	<i>JÚNIOR, PLENO E SENIOR</i>
19	<i>FRONT OU BACK END?</i>
21	<i>DESISTIR?</i>

INTRODUÇÃO



Seja bem-vindo a este eBook, onde vou concentrar meus anos de experiência para lhe dar uma outra perspectiva sobre a vida de ser um programador

Primeiramente gostaria de me apresentar: sou **Matheus Battisti**, desenvolvedor fullstack há mais de cinco anos atuando no mercado

Já trabalhei com diversas tecnologias e projetos de áreas diferentes, tive desafios desde infraestrutura até arquitetura de sistemas

Como todo iniciante passei por diversos problemas, tive dúvidas no começo da minha carreira, e por isso resolvi fazer este eBook, para lhe dar o caminho das pedras

Ensinar como se tornar um ótimo programador em menos tempo que você levaria se tivesse que descobrir tudo isso sozinho

INTRODUÇÃO



Claro que é impossível inserir toda minha experiência em um eBook

Mas alguns conceitos chaves, ou seja, os dez princípios é perfeitamente possível

E o que são os **dez princípios**?

São ensinamentos que independente da linguagem vão lhe levar a evolução

Um conhecimento atemporal que ajudará na sua caminhada

Uma quebra de paradigmas que nem todos estão dispostos a ensinar

E talvez, algumas verdades que você *não vai gostar de ler*, mas farão todo sentido e são de coração

Escritos por alguém que quer ajudar a desenvolver melhores programadores, bom proveito!

ESTOU PRONTO PARA SER UM DESENVOLVEDOR?



Esta é sem dúvida a pergunta que mais é feita entre os iniciantes

Como saber se eu estou pronto?

E a resposta é mais simples ainda, você estava pronto para andar de bicicleta quando andou pela primeira vez?

Então, porque tanto medo para ser um programador?

É exatamente o mesmo processo, você deve tentar, experimentar, cair e levantar

As empresas estarão cientes que você não tem experiência de mercado

Elas farão a aposta, porém precisam que você ofereça o seu melhor

Difícilmente alguém será demitido por um erro ou falta de experiência

Após o primeiro passo você verá que tudo se torna mais fácil

ESTOU PRONTO PARA SER UM DESENVOLVEDOR?



Meus conselhos para que você não se frustre são:

1 - Tenha um conhecimento sólido em alguma linguagem;

2 - Crie projetos para praticar o que você aprendeu;

3 - Não se limite ao front se você escolheu esta área e vice-versa;

4 - Não dependa de frameworks e libs para tudo;

5 - Nunca pare de aprender;

Se você pratica estes cinco passos, sim, **você está pronto**

A MELHOR LINGUAGEM DE PROGRAMAÇÃO



A melhor linguagem de todas, que você deve aprender agora mesmo, largar tudo e usar somente ela para todas as coisas é: **não existe**

Sei que pode doer em alguns, mas
infelizmente é a verdade

Linguagens são ferramentas, como um martelo
ou uma serra

Você corta uma madeira com um martelo?

Então, conseqüentemente, você utiliza cada
linguagem para o seu fim específico

É assim que é e é assim que vai ser

O que você pode fazer é escolher uma
linguagem para se especializar, mas sem
deixar de olhar para as outras

O mercado de TI gira muito rápido! Se você se
cegar por uma tecnologia, amanhã pode ser
tarde demais

A MELHOR LINGUAGEM DE PROGRAMAÇÃO



Em vez de ficar falando mal do PHP nas páginas por aí, aprenda o por que dela ter sido criada

Aprenda quando optar por PHP ou JavaScript

Isso sim será produtivo e vai fazer de você um melhor programador

No meu caso, utilizo o PHP como principal linguagem no meu trabalho

Mas fora dele, aprendo muito Python, que uso para Machine Learning e AI, por que PHP não tem ferramentas boas o suficientes para este fim

Além disso, procuro aprender muito JavaScript, pois gosto da ideia de uma solução do front ao back com uma só linguagem

Enfim, é só uma ideia para que você abra a mente e acabe com essa crença de melhor linguagem

NÃO FUJA DO LINUX



O Linux é um sistema operacional free, o que quer dizer que você pode instalar ele agora mesmo no seu computador

Aconselho a todos os desenvolvedores aprenderem o quanto antes

Pois a maioria dos servidores são feitos em cima de uma alguma distribuição de Linux

Por exemplo a AWS, que é uma empresa que fornece servidores e serviços em Cloud (o que é um conhecimento bem requisitado)

As instâncias da AWS são baseadas em Linux e este serviço é muito utilizado pelas empresas que lá hospedam seus projetos

Outro detalhe importante é que você vai ter uma série de dificuldades para programar no Windows, supondo que você use ele

Várias ferramentas não são adaptadas para este SO, por exemplo o Docker, ele é penoso configurar no Windows, no entanto no Linux com apenas um comando você faz a instalação

NÃO FUJA DO LINUX



Aconselho você começar pela distribuição Ubuntu

Se você aprender o básico para ser um usuário, do terminal inclusive, já é o suficiente

Você não precisa ser um DevOps, apenas um usuário que sabe os principais comandos do terminal e como funciona a estrutura de pastas

Resumindo: aprenda Linux até conseguir se virar

Faça download de uma VM, Virtual Machine, instale uma distribuição e brinque um pouco

Você não vai se machucar, eu garanto

O PROBLEMA DOS FRAMEWORKS E LIBS



Se você é iniciante fuja de frameworks até você dominar uma linguagem

O que é dominar uma linguagem?

É você conseguir fazer exatamente o que faz com um framework, mas com o seu próprio código

Por exemplo: autenticação, níveis de usuário, MVC, POO e por aí vai...

Enquanto isso não seja devoto a eles, pode lhe custar mais caro futuramente

Um bom exemplo é o do **jQuery e eu**

Aprendi jQuery antes do JavaScript e foi terrível largar ele, por causa dos seus 'benefícios'

No fundo eu só não sabia que com um pouco mais de código eu conseguiria atingir o mesmo resultado, utilizando o JS que pode ser até mais performático

O PROBLEMA DOS FRAMEWORKS E LIBS



Além disso, os frameworks e libs são bons quando você precisa de produtividade

E ruins quando você está aprendendo a programar

Você acredita que sabe fazer uma autenticação de usuários

Mas no fim usou umas 10 linhas de código e não sabe o que acontece por **baixo dos panos**

Um grande problema

Em vez de utilizar de forma irresponsável um framework, olhe o código fonte dele e aprenda mais sobre programação

E opte quando for extremamente necessário, por exemplo: quando é a ferramenta que utilizam no seu trabalho

FERRAMENTAS PARA DESENVOLVEDORES



Aconselho algumas ferramentas que vão poupar muito o seu tempo

Visual Studio Code: melhor editor de código da atualidade, experimente

MySQL Workbench: Um ótimo software para fazer consultas a banco de dados e também o esquema do banco de suas aplicações

Postman: ótima solução para testar APIs

FileZilla: software para transferência de arquivos via FTP/SFTP

GitHub: plataforma que você consegue versionar seus projetos, e ter projetos privados também de forma gratuita

Jira: software para implantação da metodologia Scrum

StackOverFlow: melhor fórum para tirar dúvidas de programação

SALÁRIO VERSUS EXPERIÊNCIA



Salário é um ponto bem crítico, mas é importante citar

A relação entre empresa e funcionário, **infelizmente**, é extremamente comercial

Ou seja, lhe pagam para fazer um serviço e é isso, na grande maioria das empresas não há emoção e afeto, apenas razão

Então, use isto a seu favor, pesquise constantemente o salário médio de sua região para a sua função, utilize o site **Glassdoor**

Obs: constantemente = 1x ao mês, não enlouqueça

Há chances de você estar com o salário defasado, use dados para negociar!

Busque sempre o que é melhor para você, *sem ser um mercenário*

Não se esqueça que se a empresa precisar, **ela vai lhe mandar embora**, então tenha os limites bem estabelecidos

FACULDADE É IMPORTANTE?



Eu acredito que a faculdade é importante, mas não é necessária

Conheço excelentes programadores autodidatas que nunca pisaram o pé lá

Mas **eu fiz** e defendo que **não é** necessário? E aí?

E aí que eu tracei uma estratégia de entrar por meio de um estágio, por que é muito mais fácil

Além de que algumas empresas, poucas, pedem diploma

Então eu pesei os dois lados e fiz minha decisão

Porém, já adianto, não me agregou muita coisa, apenas me auxiliou em algumas direções iniciais

FACULDADE É IMPORTANTE?



Quem me ajudou mesmo foi um experiente programador, que além de ser meu amigo, foi aluno da mesma faculdade

Ele me guiou por uma stack, conjunto de linguagens e ferramentas, que é bem utilizada aqui na minha região

Não deu outra, os recrutadores/gerentes nunca perguntaram nada de faculdade e rapidamente consegui um estágio

Somente perguntavam o que eu sabia, que era exatamente o que eles precisavam

Então pense e faça a **sua** escolha

VOCÊ É JÚNIOR, PLENO OU SENIOR?



Você não sabe definir?

Eu lhe explico: não tem 100% a ver com tempo de carreira, embora possa ajudar

Quantas vezes você já viu aquele seniorzão que não sabe por uma div do lado da outra? Então...

A grande métrica número um é: capacidade de resolver problemas

Quanto mais desafios você resolve, mais a sua senioridade aumenta

Fazer os deploys, análise de código, dimensionamento de infra e regra de negócio complexa são exemplos

Ou seja, a sua capacidade de transformar solicitações de outras pessoas em código

O seu código atinge sempre os requisitos das tarefas?

VOCÊ É JÚNIOR, PLENO OU SENIOR?



É por este lado que você começa a separar quem é Júnior, Pleno e Senior

(Que para mim, o título só serve para inflar ego, eu sempre digo que sou apenas um programador)

Porém, na hora de pegar aquele projeto como lead developer, essa sua capacidade e até o tempo de carreira serão avaliados, é claro

A grande métrica número dois: A sua capacidade de resolver os problemas sem solicitar ajuda

Isso mesmo, você se vira só com o **StackOverflow** e o **Google**?

Não pede ajuda para aquelas dúvidas idiotas que encontra até em português? Parabéns! Você está no caminho certo

Lembrando que o trabalho em equipe é necessário, estou falando de você *importunar* os outros devs, perceba a diferença

FRONT OU BACK END?



Você escolhe!

Mas atente-se para o outro lado também, se você tiver conhecimento das duas áreas poderá resolver mais problemas

Consequentemente ser um programador melhor e talvez ganhar mais

Não estou falando de conhecimento profundo, e sim de saber como as coisas funcionam

Como funciona o MVC? Como deixo um site responsivo?

Pela perspectiva de uma área ou outra, podem ser perguntas super complexas

Mas entender a lógica dos dois problemas e saber como são resolvidos, é bem simples

Utilize seu tempo de estudo para pisar no outro lado

FRONT OU BACK END?



Já eu, sempre tive curiosidade em ambos os lados

Nunca fiquei contente em trabalhar em uma das partes somente

Sempre quis fazer a solução por completo e é bem divertido

Ao contrário do que os outros dizem, que você vai saber apenas um pouco de tudo e não será bom em nada

Quem escolhe o quanto vai conhecer de algum assunto, é apenas você

Meu conselho: aprenda um pouco de tudo, esse conhecimento só vai lhe ajudar um programador melhor

QUANDO DEVO DESISTIR?



Nem pense nisso, o segredo é estudar bastante e persistir

Como em todas as outras áreas, porém com vantagens

Você tem materiais gratuitos sobre tudo na internet

Quer uma dica para aprender melhor?

Aprenda inglês, vai mudar toda a sua perspectiva sobre o assunto

Os melhores conteúdos são em inglês, isso é fato

Se ainda não estiver convencido, explore mais sobre a linguagem que escolheu, não fique mudando só porque encontrou uma dificuldade

Tente solucionar algum problema que você tem na sua casa, programar um organizador de finanças, por exemplo

QUANDO DEVO DESISTIR?



Ou melhor, copie um software que já existe no mercado

Crie um clone do Twitter, apenas com a regra de negócio principal

Esses projetos são os que mais abrem a mente, você **precisa praticar**

Programação é para qualquer um, não é questão de dom e sim de esforço :)



Obrigado

Enquanto houver pessoas dispostas
a aprender, eu estarei criando
conteúdo para enriquecer ainda
mais os seus conhecimentos

Matheus Battisti

Hora de Codar