

Disclaimer



- Sem ver a palestra toda estes slides são como uma arma na mão de uma criança. Eles (nem todas minhas palestras são assim) foram propositalmente criados para serem provocativos e só quem viu a palestra entenderá o contexto, percebendo as contradições entre o slide e o que eu falo brincando com a situação, por isso não me responsabilizo pelo mau julgamento do conteúdo!
- Me desculpe, minhas palestras são muito pessoais, o uso por outras pessoas certamente implicará em entendimento equivocada da intenção dela.
- Estou à disposição para apresentar em outro eventos e empresas.



THE DEVELOPER'S CONFERENCE

Design de código Performance é feature

Antonio Maniero
Microsoft MVP

Antonio Maniero



THE
DEVELOPER'S
CONFERENCE

- ❖ Engenheiro desde que nasceu, escovador de bits desde o primeiro computador c/ 2KB de memória e tarado por eficiência e simplicidade
- ❖ Desenvolvedor generalista há 35 anos. Especialista em ERP. Sistemas LOB rodando em dezenas de milhares de empresas de todos os portes e setores no Brasil e exterior para milhões de usuários corporativos
- ❖ Aficionado por linguagens de programação
- ❖ Apaixonado por ensinar, compartilhar conhecimento
- ❖ MVP sem querer 😊



pt.stackoverflow.com/users/101/maniero



linkedin.com/in/maniero/



facebook.com/antonio.maniero.junior



twitter@manieromvp



Manual da palestra



THE
DEVELOPER'S
CONFERENCE

- É uma degustação, depois posso oferecer mais
- Não quero vender nada para você
- Fundamentação é meu foco, mas não cabe aqui
- Pensamento fora da caixa, contestação do status quo
- Serei assertivo, mas isso não quer dizer que sei tudo
- Usarei hipérboles, simplificações e generalizações
- Prioridade para performance
- Analise seu contexto



Estado da nossa indústria



THE
DEVELOPER'S
CONFERENCE

- Tudo é para ontem
- Eterno MVP
- Discurso de modernidade
- “Adoção” de tecnologias por marketing
- Muita informação, pouca profundidade
- Cultura das Dicas Miojo® e das “boas práticas”
- Visão de curto prazo
- Nova solução para o problema causado pela solução anterior



Otimização



THE
DEVELOPER'S
CONFERENCE



Donald Knuth

“Os programadores gastam muito tempo pensando ou se preocupando com a velocidade de partes não-críticas de seus programas, e essas tentativas de eficiência realmente têm um forte impacto negativo quando a depuração e a manutenção são consideradas. Devemos esquecer as pequenas eficiências, digamos 97% do tempo: a otimização prematura é a raiz de todo o mal. No entanto, não devemos deixar passar nossas oportunidades nessa crítica de 3%”

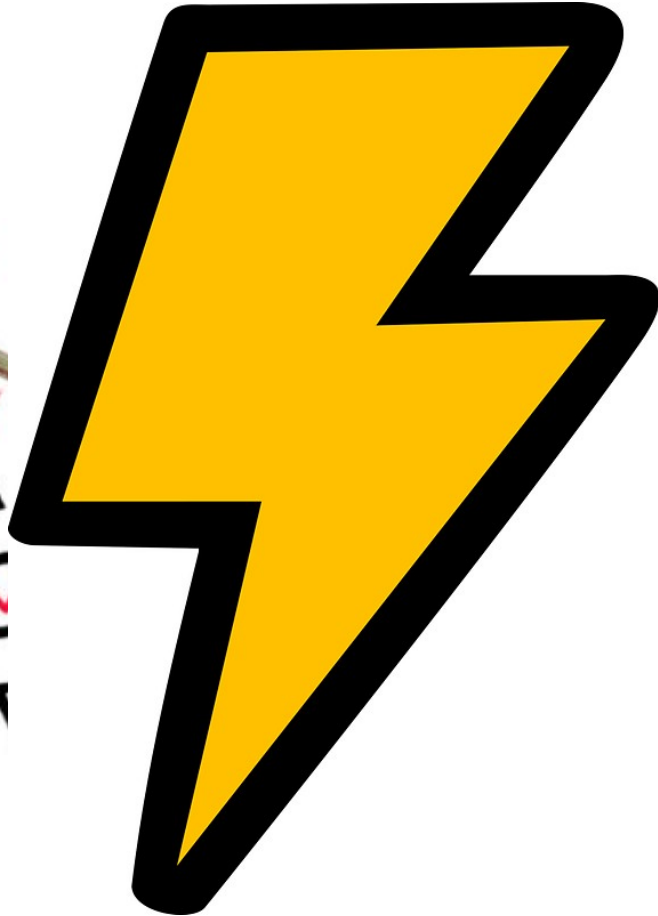
Funcionar é diferente de estar
certo



THE
DEVELOPER'S
CONFERENCE



Requisitos



THE
DEVELOPER'S
CONFERENCE

“ *Boas especificações sempre melhorarão a produtividade do programador muito mais que qualquer técnica ou ferramenta de programação.* ”

Milt Bryce

Casos reais



stackoverflow



THE
DEVELOPER'S
CONFERENCE

209,420,973 HTTP requests to our load balancer

66,294,789 of those were page loads

1,240,266,346,053 bytes (1.24 TB) of HTTP traffic sent

569,449,470,023 bytes (569 GB) **total received**

3,084,303,599,266 bytes (3.08 TB) total sent

504,816,843 SQL Queries (from HTTP requests alone)

5,831,683,114 Redis hits

17,158,874 Elastic searches

3,661,134 Tag Engine requests

607,073,066 ms (168 hours) spent running SQL queries

10,396,073 ms (2.8 hours) spent on Redis hits

147,018,571 ms (40.8 hours) spent on Tag Engine requests

1,609,944,301 ms (447 hours) spent processing in ASP.Net

22.71 ms average (19.12 ms in ASP.Net) for 49,180,275 question page renders

11.80 ms average (8.81 ms in ASP.Net) for 6,370,076 home page renders

Casos reais



stack overflow



THE DEVELOPER'S CONFERENCE

● NY-WEB01	7 %		<u>44.60 GB / 63.91 GB (69.79%)</u>
● NY-WEB02	4 %		<u>48.64 GB / 63.91 GB (76.11%)</u>
● NY-WEB03	3 %		<u>44.22 GB / 63.91 GB (69.19%)</u>
● NY-WEB04	3 %		<u>43.44 GB / 63.91 GB (67.98%)</u>
● NY-WEB05	9 %		<u>48.59 GB / 63.91 GB (76.04%)</u>
● NY-WEB06	8 %		<u>47.63 GB / 63.91 GB (74.53%)</u>
● NY-WEB07	4 %		<u>38.11 GB / 63.91 GB (59.64%)</u>
● NY-WEB08	8 %		<u>45.91 GB / 63.91 GB (71.84%)</u>
● NY-WEB09	5 %		<u>41.34 GB / 63.91 GB (64.69%)</u>
● NY-WEB10	0 %		<u>28.35 GB / 63.91 GB (44.35%)</u>
● NY-WEB11	0 %		<u>35.16 GB / 63.91 GB (55.01%)</u>
● NY-WEB12	1 %		<u>5.16 GB / 64.00 GB (8.06%)</u>

Node	CPU	Memory
● CO-SQL01	0 %	<u>365.12 GB / 383.96 GB (95.09%)</u>
● CO-SQL02	1 %	<u>474.10 GB / 767.91 GB (61.74%)</u>
● NY-SQL01	0 %	<u>75.57 GB / 383.96 GB (19.68%)</u>
● NY-SQL02	6 %	<u>365.18 GB / 383.96 GB (95.11%)</u>
● NY-SQL03	1 %	<u>94.76 GB / 767.90 GB (12.34%)</u>
● NY-SQL04	13 %	<u>731.53 GB / 767.90 GB (95.26%)</u>

Casos reais



stack overflow



THE
DEVELOPER'S
CONFERENCE

Qual o segredo?

Pragmatismo

Eficiência

Casos reais



stackoverflow



THE
DEVELOPER'S
CONFERENCE

Por que fazem isto?

Custo

Experiência do usuário (SEO)

Reinventando a roda



THE
DEVELOPER'S
CONFERENCE



Reinventando a roda

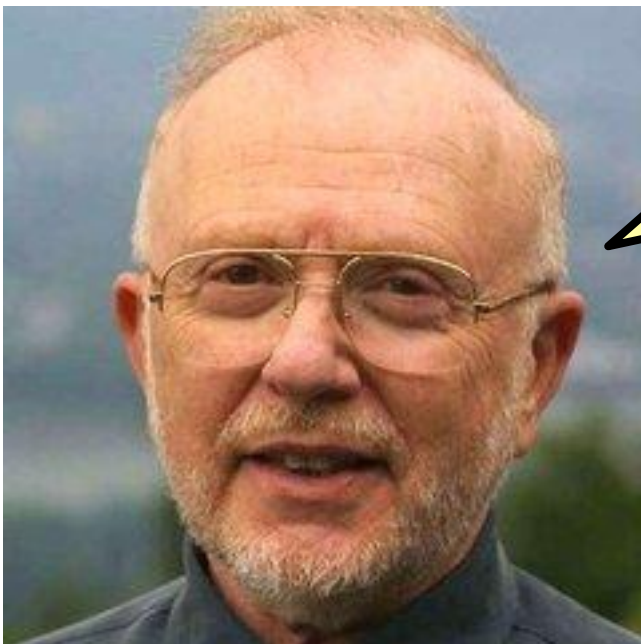


THE
DEVELOPER'S
CONFERENCE





THE
DEVELOPER'S
CONFERENCE



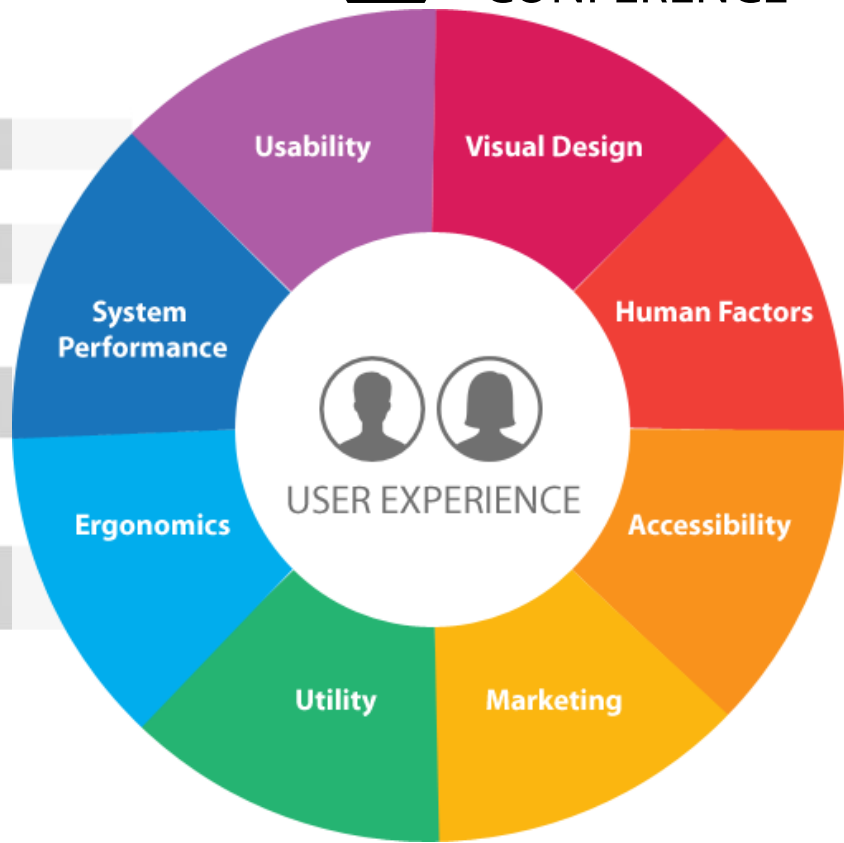
“Um mau programador pode facilmente criar dois novos empregos por ano.”

David Parnas

Performance não é ser rápido



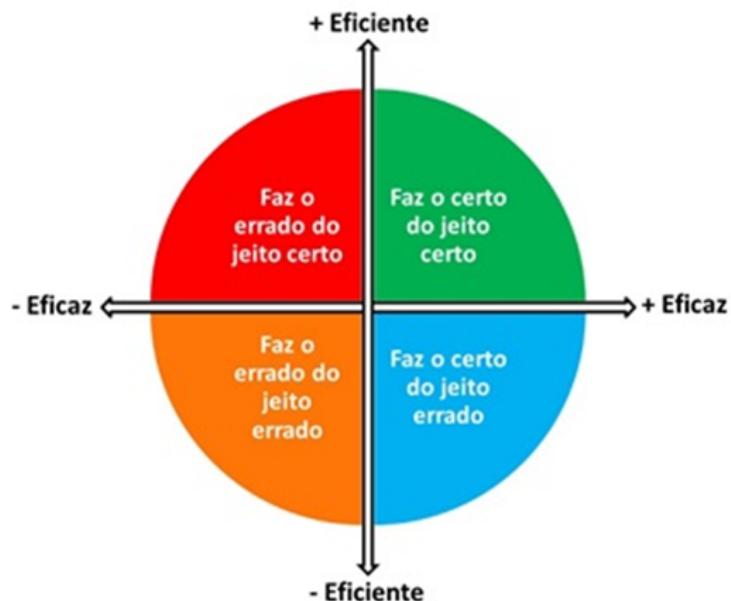
THE
DEVELOPER'S
CONFERENCE



Performance é ser eficiente



THE
DEVELOPER'S
CONFERENCE



EFICIÊNCIA

CUSTO

Fazer corretamente

Utilizar produtivamente os recursos

Custo-benefício

Mínimo de perdas e/ou desperdícios

EFICÁCIA

RESULTADO

Fazer o que deve ser feito

Capacidade de atingir objetivos

Cumprir metas

Realizar o que foi proposto

EFETIVIDADE

IMPACTO

Fazer corretamente o que tem que ser feito

Transformar a situação existente

Mudança e desenvolvimento

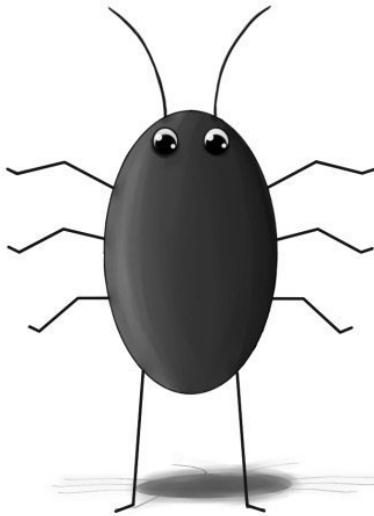
Relação entre a produção e capacidade de produtos

Me engana que eu gosto

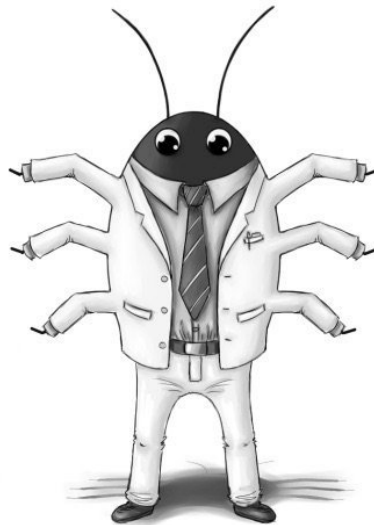


THE
DEVELOPER'S
CONFERENCE

iris-hime.deviantart.com



BUG



FEATURE



BY DESIGN

Linguagens mais rápidas?



Assembly ?

C ?

C++ ?

C# ?

Java ?



Por que não?

C# versus C++

Rico Mariani



09 de maio de 2005

Chinese/English dictionary



<https://blogs.msdn.microsoft.com/ricom/>



Raymond Chen



<https://blogs.msdn.microsoft.com/oldnewthing/>

<https://blogs.msdn.microsoft.com/ricom/2005/05/10/performance-quiz-6-chineseenglish-dictionary-reader/>

C# versus C++



- Uma tradução linha-a-linha do código C++ original em C# foi executada 10 vezes mais rápido que o código C++
- Foram necessárias cinco otimizações diferentes (uma delas introduziu um bug) para que o código C++ correspondesse à velocidade do código C# inalterado
- Após a sexta otimização de Raymond, seu código C++ finalmente venceu o código C# - porque o tempo de execução foi reduzido para onde a sobrecarga de inicialização de 60ms do CLR fez a diferença!
 - Para conseguir isso, Raymond teve que escrever seu próprio:
 - Sistema de arquivo/io
 - Alocador
 - Mapeamento internacional

Ferramenta certa



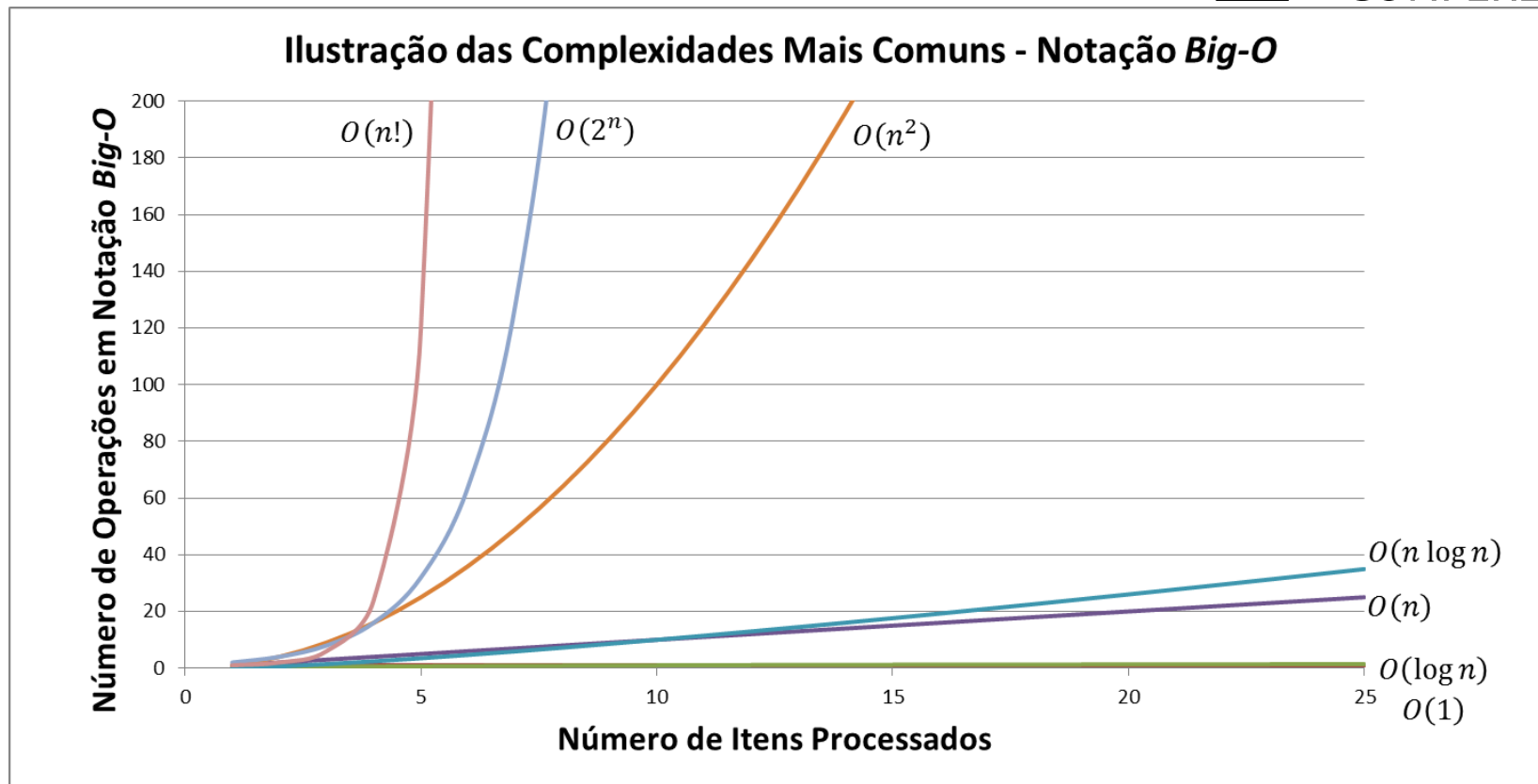
THE
DEVELOPER'S
CONFERENCE



Ferramenta certa



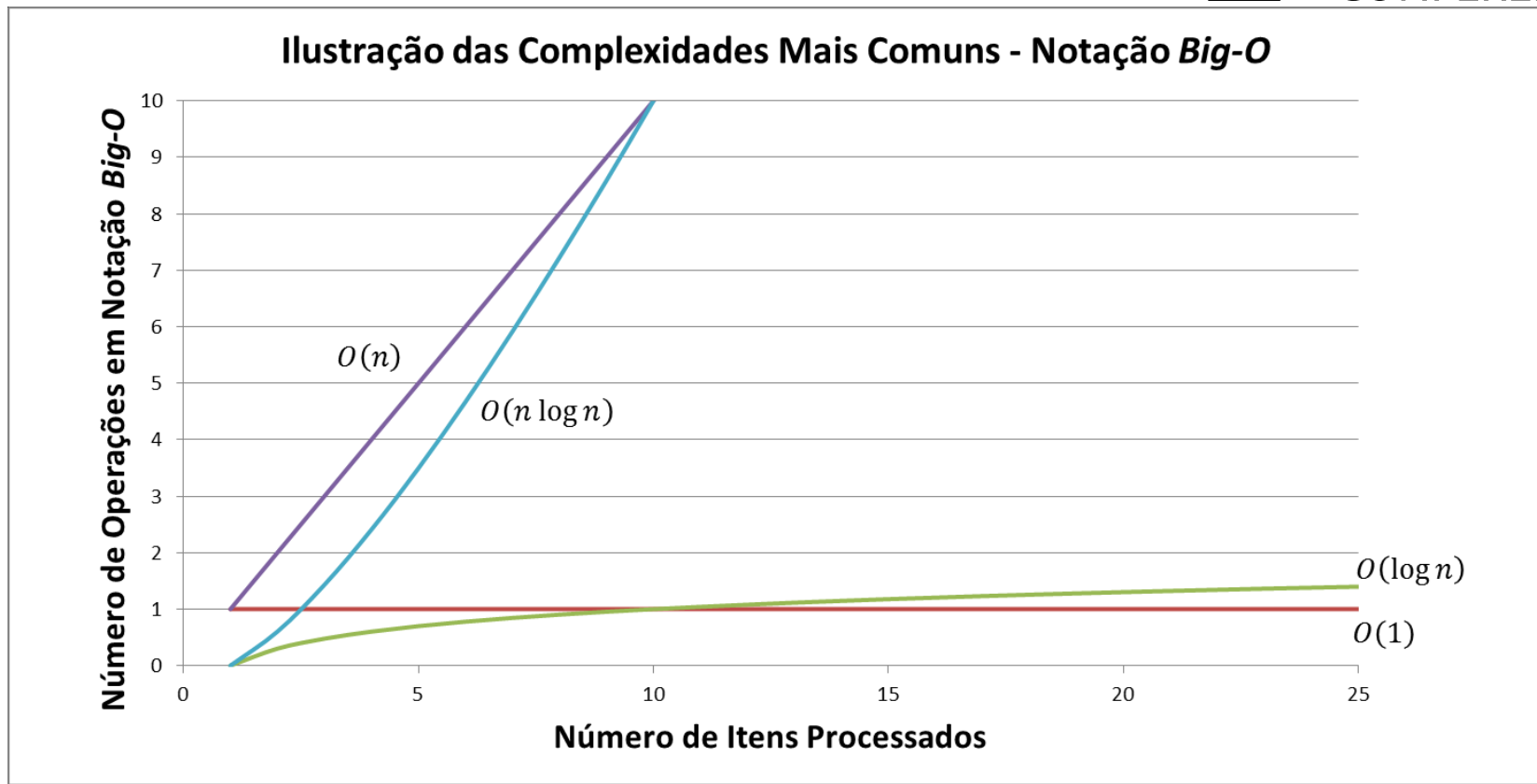
THE
DEVELOPER'S
CONFERENCE



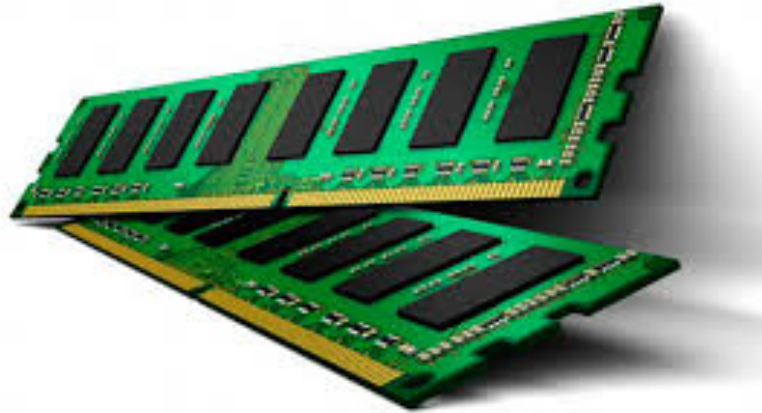
Ferramenta certa



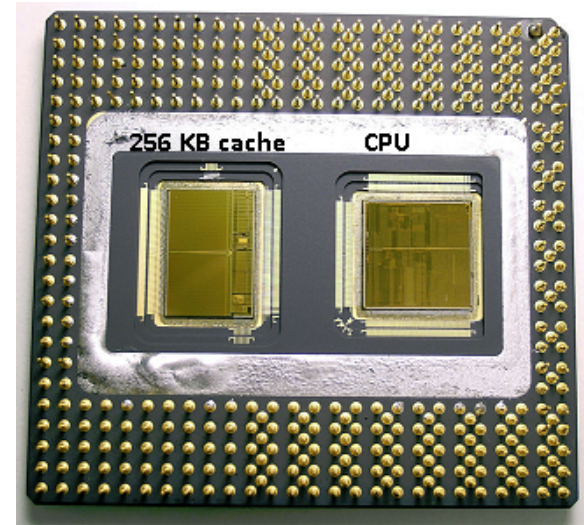
THE
DEVELOPER'S
CONFERENCE



Memória é barata?



8MB ~ US\$ 350



Memória é barata?



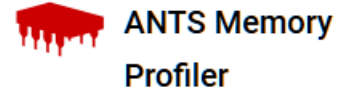
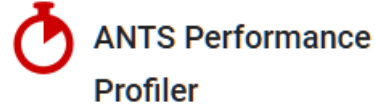
THE
DEVELOPER'S
CONFERENCE

Meio	Tamanho típico	Tempo de acesso
Virtual Memory	1GB – 16GB	50 – 100 us
Physical Memory	2GB – 1TB	60 – 150 ns
3nd-Level Cache	2MB – 16MB	25 – 40 ns
2nd-Level Cache	128KB – 2MB	6 – 12 ns
1st-Level Cache	16KB – 128KB	1 – 3 ns
Registers	16 – 64 Words	0.2 – 0.5 ns

Profilers



ReSharper



.NET Memory Profiler



Escolha seu veneno



THE
DEVELOPER'S
CONFERENCE

/
if
for
IO ingênuo
Debug
new List()
Add()
try-catch
LINQ/Stream
Abstração
Parâmetros



>>
switch
for each
4KB
Release
New List(1_000_000)
AddRange()
Evite (inlining)
For each
Concreto
Objetos

Escolha seu veneno



Síncrono

Async

Thread

Lock

Alocação desmedida

Heap

No cache

Eager/Lazy

Exceção



Async

Síncrono

Execução serial

Imutável

Economia

Stack

Cache

Lazy/Eager

Retorne erro

Escolha seu veneno



Linked List

Dictionary

$O()$**

Build Strings

Array

double

Regex

Serialização

32 bits



Tree based

Array

$O(1)$

StringBuilder

SIMD

decimal

Análise manual

Acesso direto

64 bits

Escolha seu veneno



ORM

Db poderoso

SQL

NoSQL

Framework

Microsserviços

SOAP

RESTy

TCP

Otimização



MicroORM/No ORM

Db leve

NoSQL

SQL

Library

Monolítico

RESTy

TCP/UDP

IPC (Pipes/MMap)

Perfilamento

Escolha seu veneno



Herança

Encapsulamento

virtual

Reflexão

Runtime

Patterns

Domain Drive Dev.



Composição

Acesso direto

Non-virtual/static

Gerador de código

Compile time

Library/Scaffolding

Data Driven Develop.

OOP

Você sabe o que
está fazendo



Microserviços

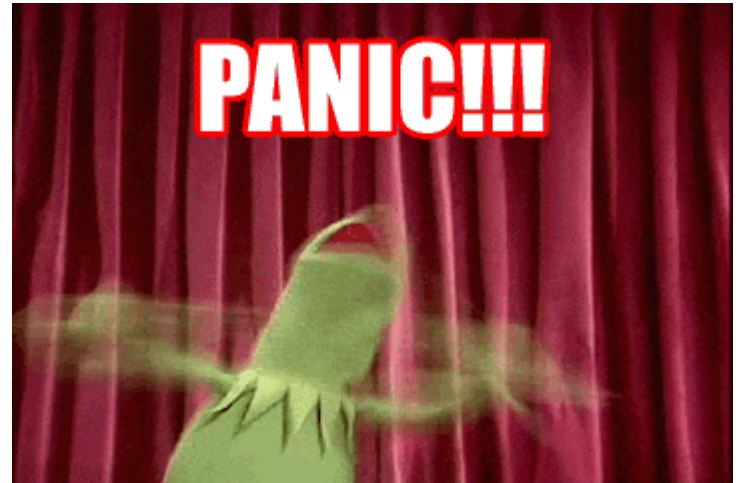
Você sabe onde
está se metendo



DDD



Vai tonto!



Desenvolver é complicado

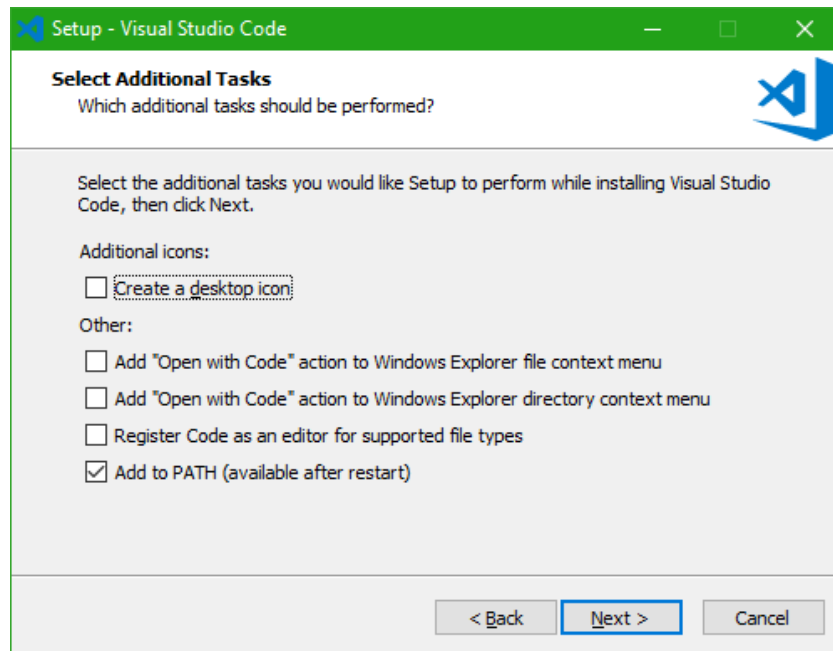
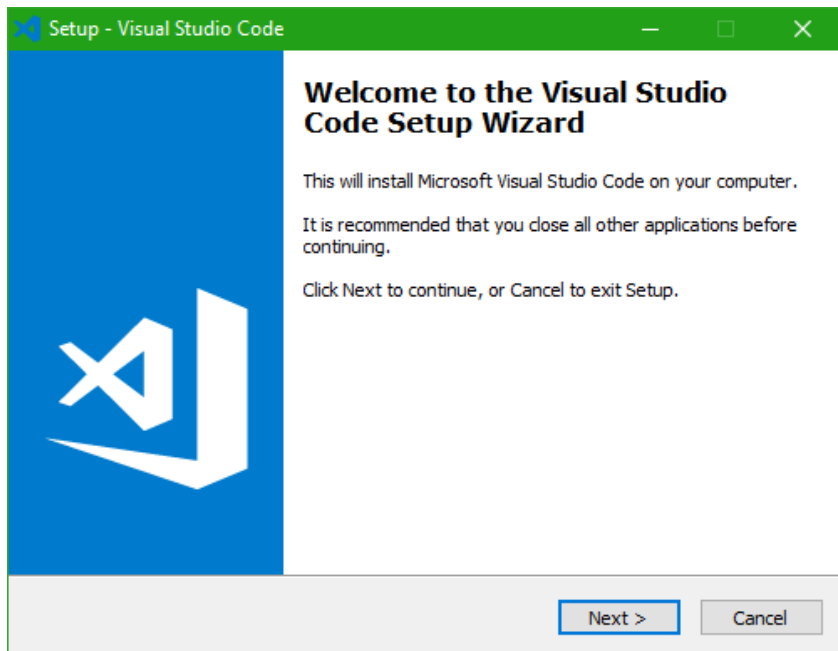


THE
DEVELOPER'S
CONFERENCE

Não podemos
achar que é só
juntar as pecinhas
de Lego e tudo
será lindo.



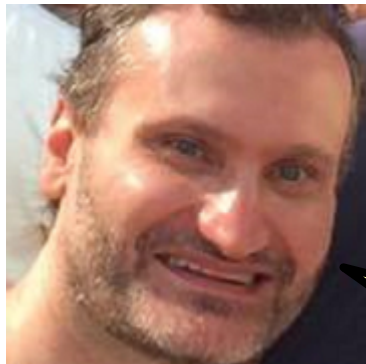
Wizard programming



Entenda o que está fazendo



THE
DEVELOPER'S
CONFERENCE



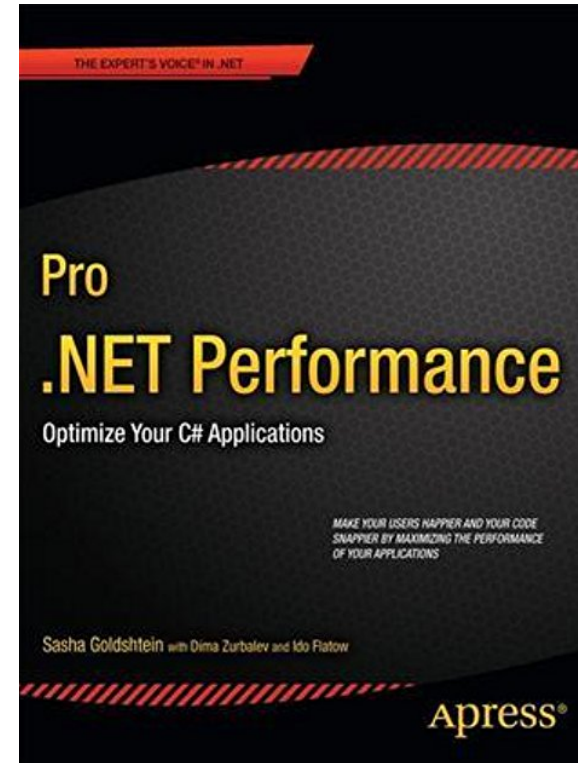
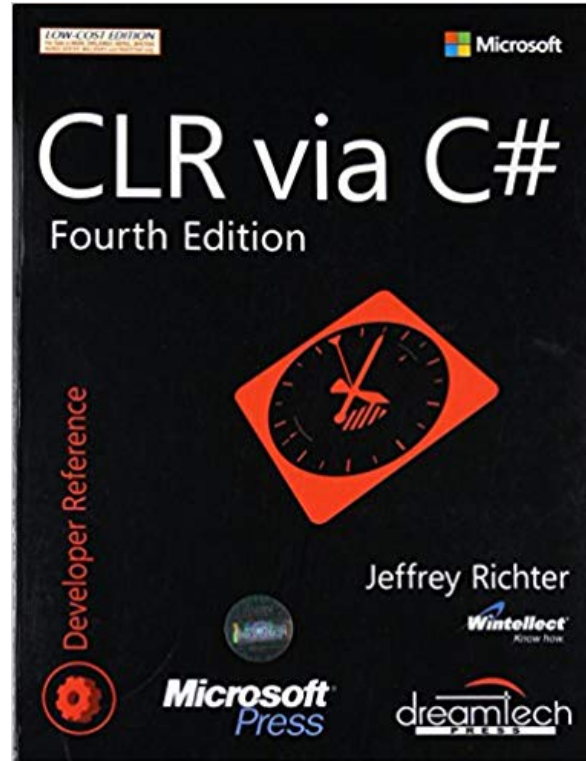
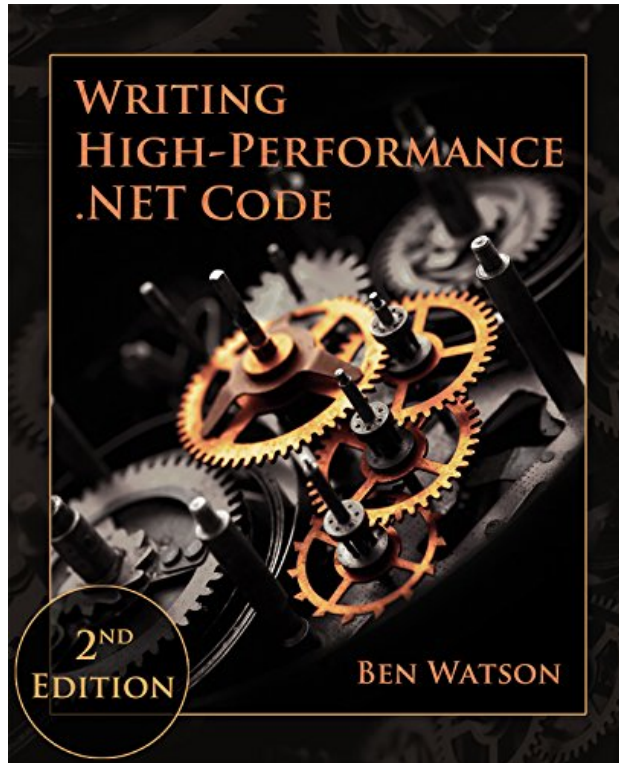
Antonio Maniero

“Enquanto você não souber o que cada caractere do seu código faz, até mesmo o espaço em branco, você ainda não sabe programar”

Sugestões



THE
DEVELOPER'S
CONFERENCE



Ainda não acabou!



THE
DEVELOPER'S
CONFERENCE

- Não acredite em tudo que chega até você
 - Incluindo esta palestra
 - Questione tudo! Mas não entre em paranoia 😊
 - Procure visões diferentes
 - Uma “mentira” repetida mil vezes não a faz verdadeira
 - Só porque deu certo para alguém não significa que serve para você
 - Atenção ao **contexto!**



THE
DEVELOPER'S
CONFERENCE

➤ Contexto

➤ Contexto

➤ Contexto

Não encerramos aqui



THE
DEVELOPER'S
CONFERENCE

- Me procurem, me sigam, me adicionem, perguntem, interajam, façam networking
- Gostaram? Querem mais?
 - Me convidem para seus eventos
 - Peçam uma palestra minha no seu evento preferido

Obrigado

Antonio Maniero



<https://about.me/maniero>





THE DEVELOPER'S CONFERENCE