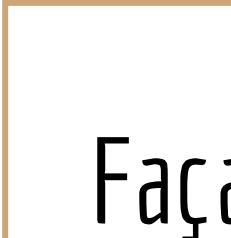



#cpbr12



Faça Muito Mais  
com Ciência de  
Dados, C# E  
Rust



# Sumário

- Agilidade com C#
- Inteligência com Data Science
- Rapidez e Segurança com Rust

Agilidade com C#

# Agilidade

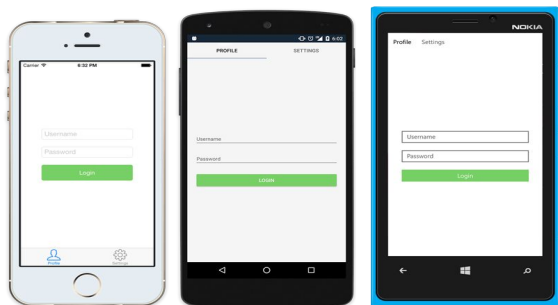


# C#



- C# (C-like Object Oriented Language)
- Nasce dentro da plataforma .Net, nova estratégia da empresa
- Foco em produtividade, desempenho e unificação de produtos

# Aplicações



# Ferramentas

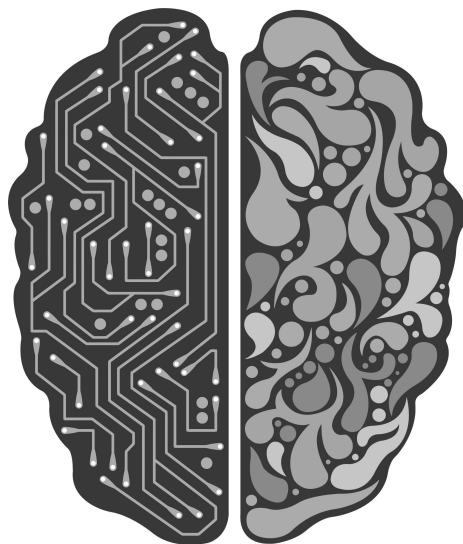
- Visual Studio
- VS Code
- Git
- Banco de Dados (ex: Sql Server)

# Vantagens

- Criação rápida de protótipos e produtos de qualidade
- Fail Fast
- Plataforma com suporte e ecossistema rico
- Grande quantidade de material disponível



# Inteligência com Data Science



# Uma Breve Contextualização

- Barateamento de hardware
- Desenvolvimento de tecnologias
- Hardware e software mais eficientes
- Crescimento constante na capacidade de armazenamento de dados
- Invenção de novos dispositivos capazes de gerar hardware



DATA

# Data Scientist: The Sexiest Job of the 21st Century

by **Thomas H. Davenport** and **D.J. Patil**

FROM THE OCTOBER 2012 ISSUE



SUMMARY



SAVE



SHARE



COMMENT



TEXT SIZE



PRINT

**\$8.95** BUY COPIES

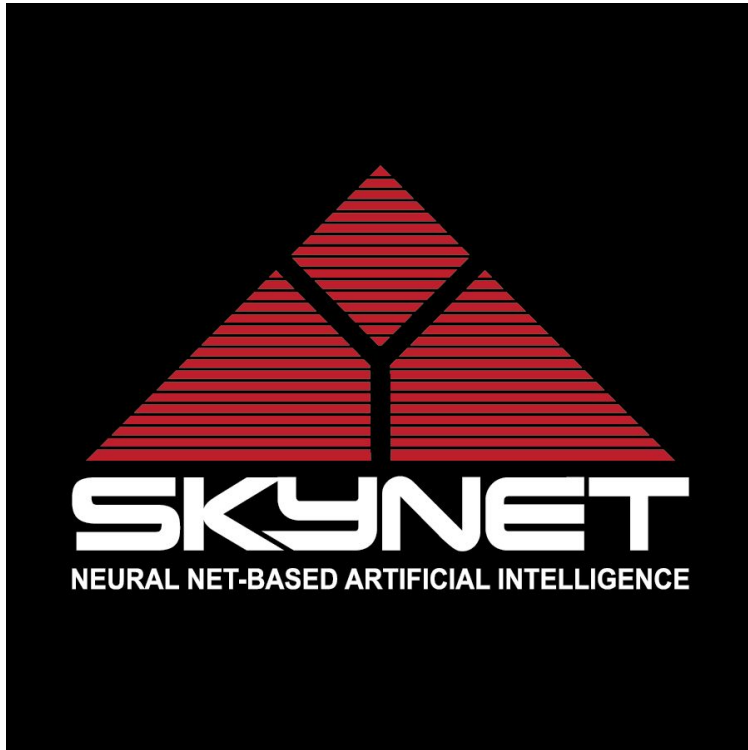
**W**hen Jonathan Goldman arrived for work in June 2006 at LinkedIn, the business networking site, the place still felt like a start-up. The company had just under 8 million accounts, and the number was growing quickly as existing members invited their friends and colleagues to join. But users weren't seeking out connections with the people who were already on the site at the rate executives had expected. Something was apparently missing in the social experience. As one LinkedIn manager put it, "It was like arriving at a conference reception and realizing you don't know anyone. So you just stand in the corner sipping your drink—and you probably leave early." Goldman, a PhD in physics from Stanford, was intrigued by the linking he did see going on and by the richness of the user profiles. It all made for messy data and unwieldy analysis,

# Ciência de Dados



- Extração de conhecimento vindo de um volume de dados
- Área interdisciplinar que envolve Matemática, Estatística, Eng de Software, Inteligência Artificial e Domínio de Negócio
- É uma ciência de fato

# Para Que Serve?



- Gerar insights através de modelos de dados que podem realizar previsões, análises, automações, etc...  
diminuindo ou abdicando a dependência do fator humano
- Criação de casas, carros, cidades, negócios inteligentes, assim como servir de ferramenta de auxílio para diversas outras áreas da ciência e da indústria.

# 0 Poder da Ciência de Dados

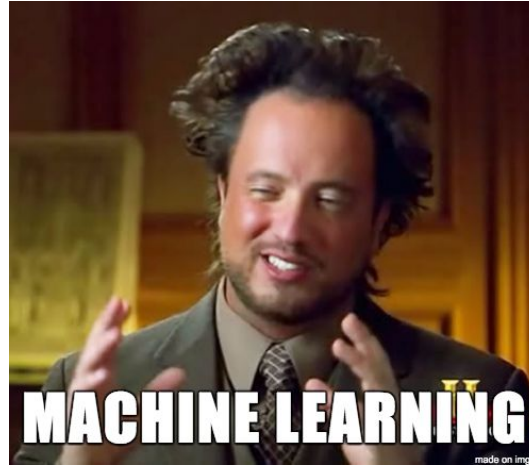
- Escândalo Cambridge Analítica
- Influência em eleições
- Google Duplex
- Sistemas de Recomendações poderosos (Spotify, Amazon, Netflix)
- Caso da loja Target
- Processamento de Linguagem Natural
- Detecção de Objetos
- Entre outros



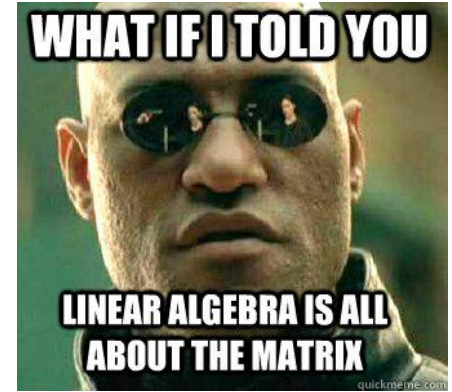
# Mão Na Massa

$$\mathbf{B} > \frac{1}{n} \sum_{i=1}^n x_i$$

Estatística



Aprendizado de Máquina



Álgebra Linear

# Ferramentas para Ciência de Dados

- Python
- Tensorflow
- Scikit Learn
- Pandas
- Matplotlib e Seaborn
- R
- Jupyter Notebook
- Tableau
- CNTK, Pytorch, etc..



python™



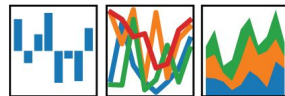
TensorFlow



Keras



pandas  
 $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$





# Afinando os Conhecimentos

The logo for Kaggle, featuring the word "kaggle" in a lowercase, blue, sans-serif font.

<https://www.kaggle.com/>

# Rapidez e Segurança com Rust

# Rust



- Jogo de sobrevivência
- Multijogador
- Primeira pessoa
- Publicado em 2018
- Inicialmente era um clone de DayZ que misturava elementos de Minecraft

# Rust



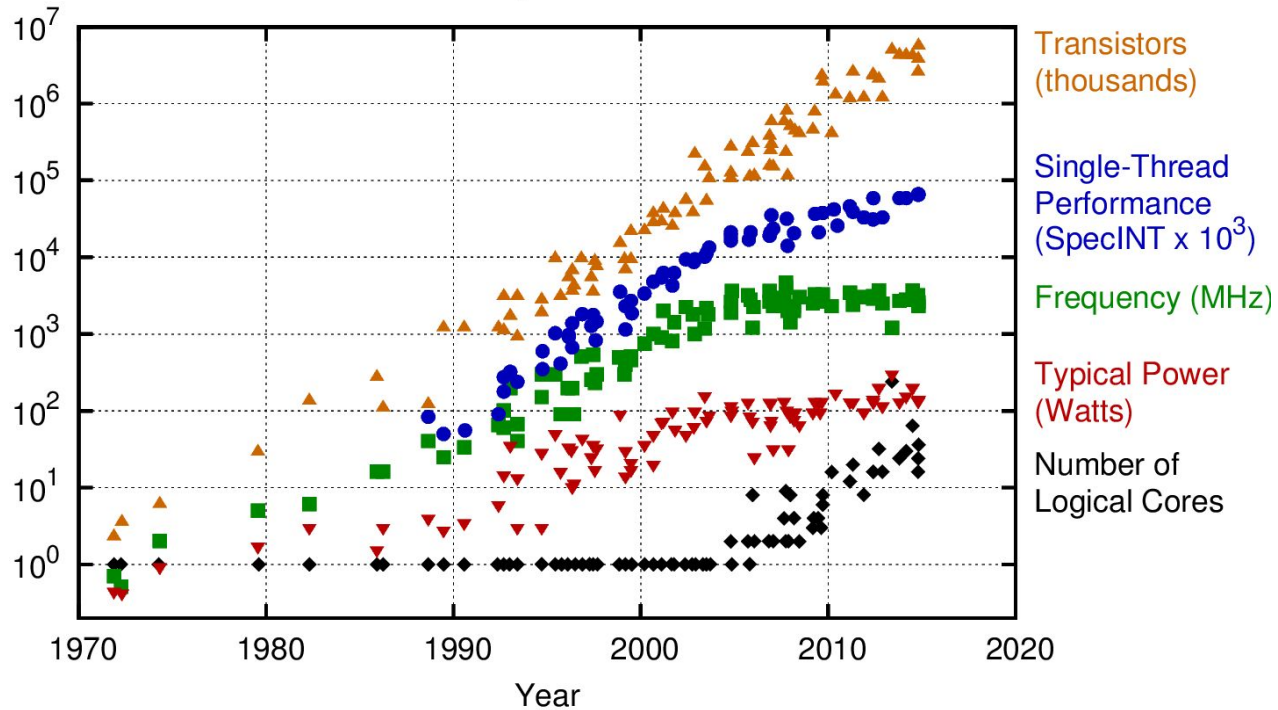
- Linguagem de programação
  - Memory-safe e thread-safe
  - Tão rápido e leve quanto C
  - Sem Garbage Collector e Sem Runtime
  - Orientada a Objetos e Funcional
- 
- Gerenciador de Pacotes, Testes, Traits e Generics

# Por que criaram Rust?

“O clock dos processadores dobra a cada 18 meses.”

Lei de Moore, 1965.

## 40 Years of Microprocessor Trend Data



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten  
New plot and data collected for 2010-2015 by K. Rupp

“The way the processor industry is going, is to add more and more cores, but nobody knows how to program those things. I mean, two, yeah; four, not really; eight, forget it.”

Steve Jobs

# Por que utilizar Rust?

**Bug 650064**

**Running Aurora and Firefox in parallel**

**UNCONFIRMED** Unassigned

▼ **Status**

Product: Firefox ▼

Reported: 8 years ago

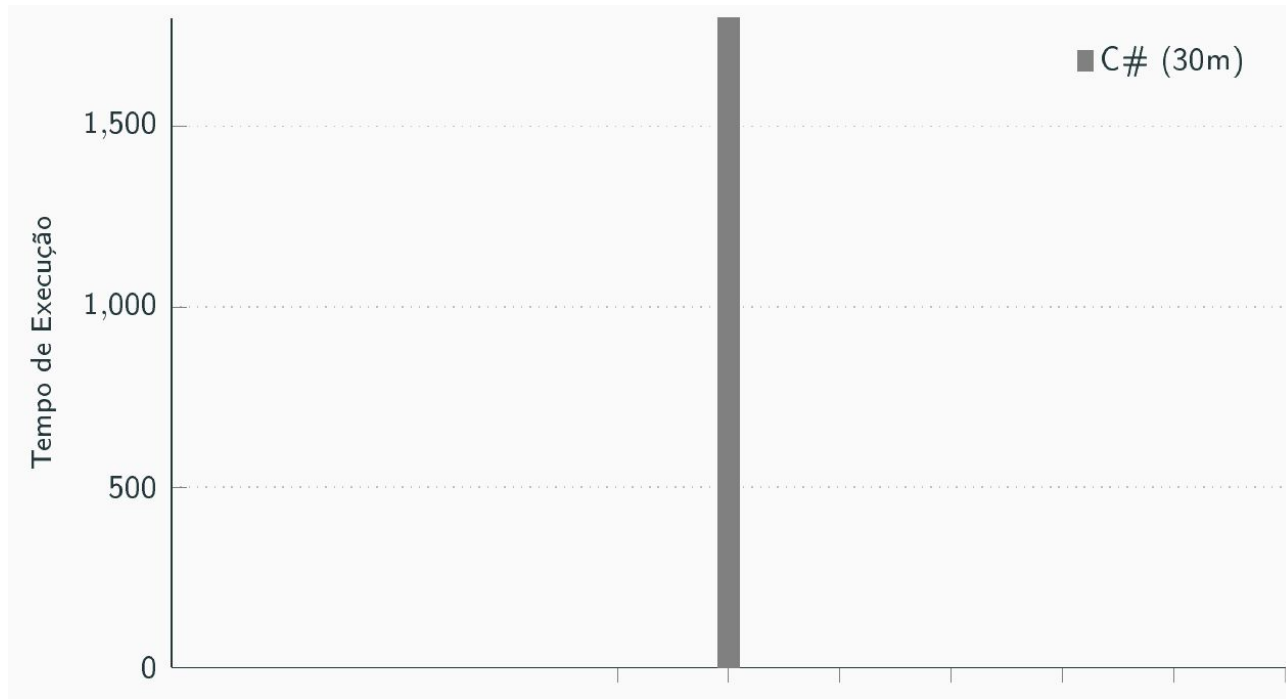
Component: General ▼

Modified: 6 years ago

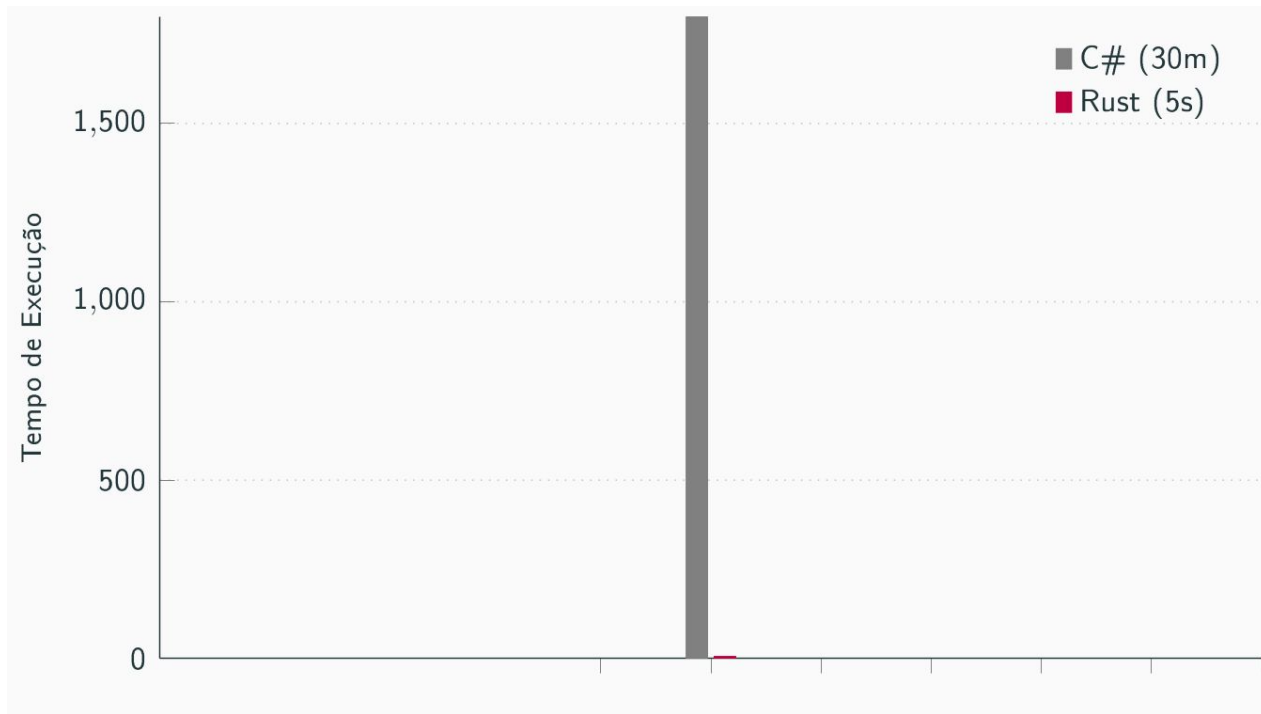
Status: UNCONFIRMED



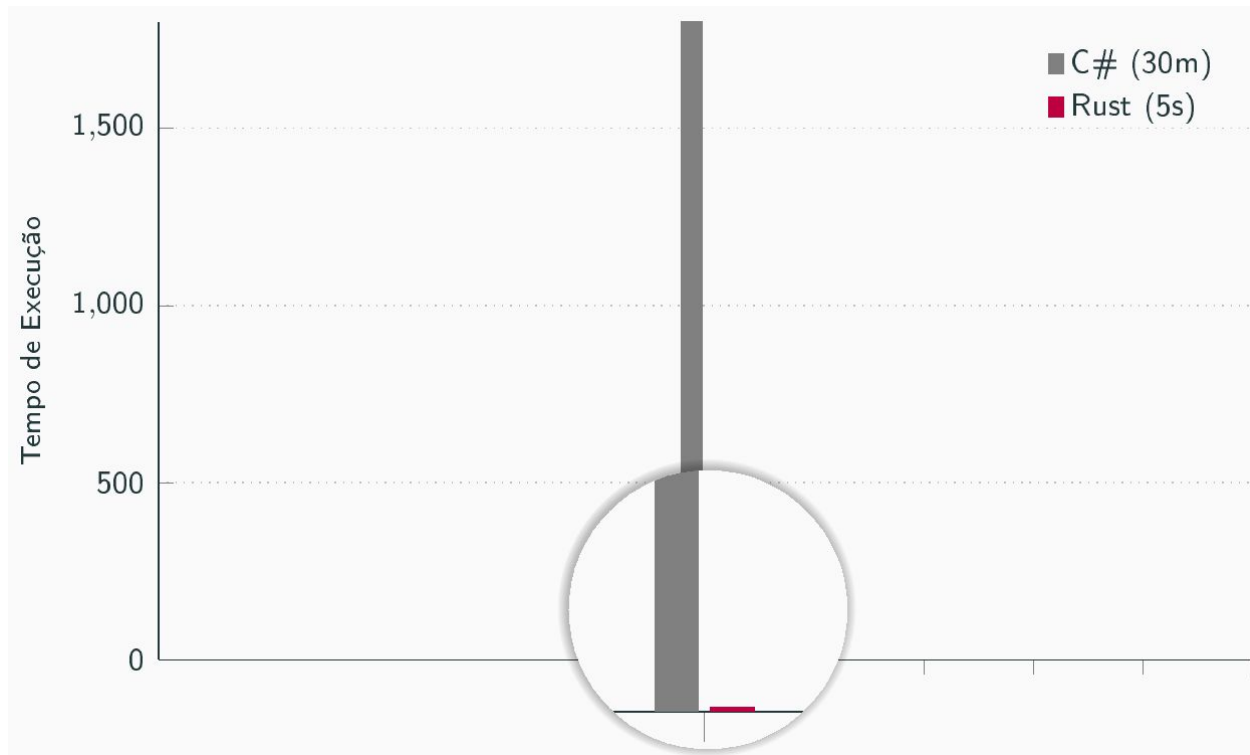
# Por que utilizar Rust?



# Por que utilizar Rust?



# Por que utilizar Rust?



Quem utiliza Rust em produção?

# Quem usa e para que?



"Optimizing cloud file-storage."

Quem usa e para que?

**moz://a**

“Building the Servo browser engine, integrating into Firefox, other projects.”

# Quem usa e para que?

The logo for Canonical, featuring the word "CANONICAL" in a bold, sans-serif font. The letter "O" is replaced by a stylized circle with a dot in the center, resembling an eye or a target.

“Everything from server monitoring to middleware.”

# Quem usa e para que?



"We use Rust in a service for analyzing petabytes of source code."



# Quem usa e para que?



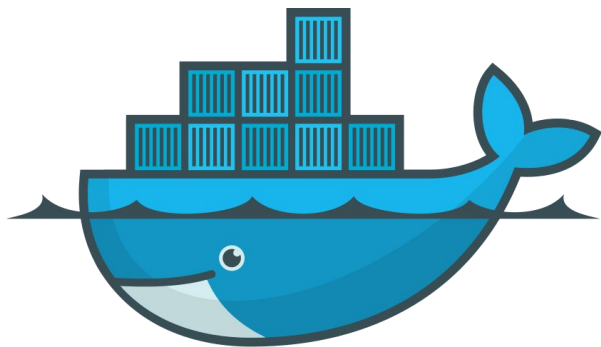
"Programming Assignments in secured Docker containers."

# Quem usa e para que?




“Replacing C and rewriting performance-critical bottlenecks in the registry service architecture.”

# Dockerizando uma aplicação





1. Compile o código em Rust utilizando uma imagem de build para a arquitetura x86\_64 Linux;
2. Copie o executável para uma imagem Scratch;
3. Defina o ponto de entrada da imagem para o executável;

# Dockerizando uma aplicação



Dashboard

PRIVATE REPOSITORY

johnfercher/ 

Last pushed: a month ago

[Repo Info](#) [Tags](#) [Collaborators](#) [Webhooks](#) [Settings](#)

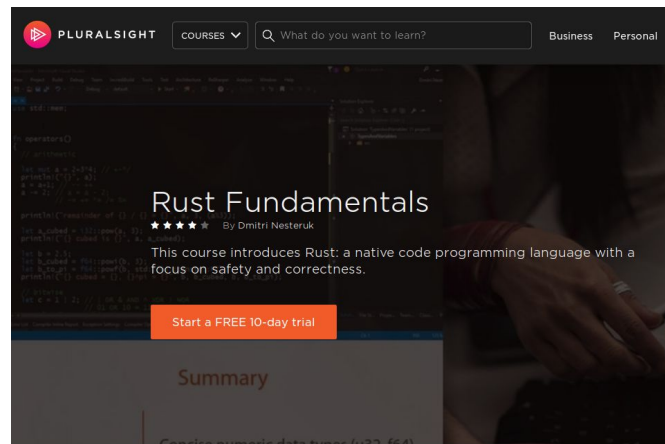
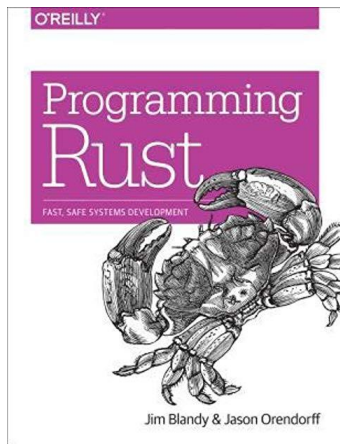
Tag Name	Compressed Size	Last Updated
latest	4 MB	a month ago

# Limitando o uso de recursos



```
let pool = ThreadPoolBuilder::new()  
    .num_threads(6)  
    .stack_size(4_000_000)  
    .build()?;  
  
pool.install(|| {  
    let primes = numbers.par_iter()  
        .filter(|x| is_prime(x))  
        .collect::});
```

# Material de Estudos



Telegram Rust Brasil: <https://t.me/rustlangbr>

# Obrigado



<https://github.com/johnfercher/cpbr12>