braspag
UMA EMPRESA CIELO

Eliminando Gargalos de Processamento Utilizando Rust

Johnathan Fercher

Sumário

Introdução

Introdução

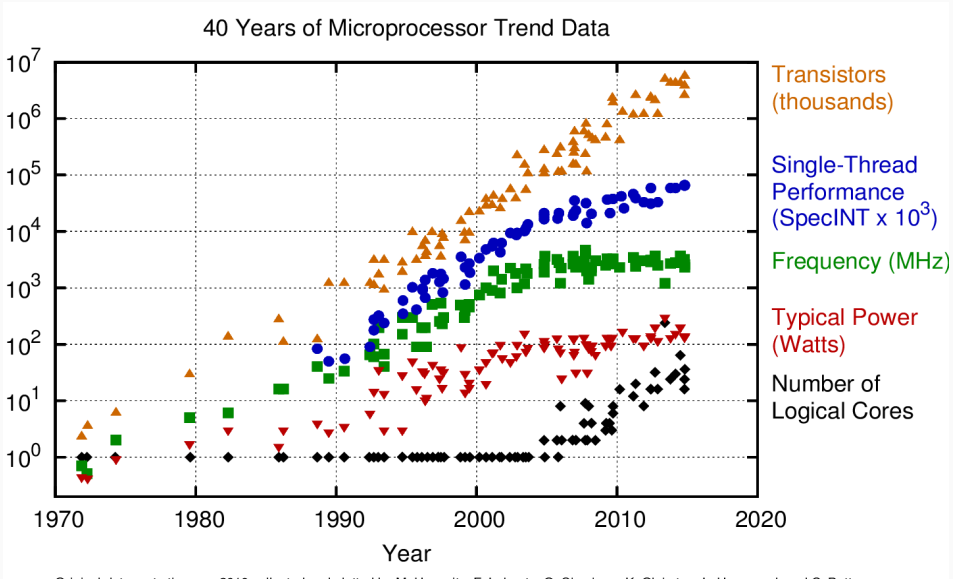
- Programming Language;
- Memory and thread-safe;
- Rust $\rightarrow$ LLVM $\rightarrow$ EXE;
- C-Bindings;
- Object-Oriented and Functional;
- Unit-tests and Package Manager;
- Interfaces and Generics;
- Without Garbage Collector;

Motivação

"O clock dos processadores dobra a cada 18 meses."

Lei de Moore, 1965.

Motivação


40 Years of Microprocessor Trend Data
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
1970
1980
1990
2000
2010
2020
Year
Transistors (thousands)
Single-Thread Performance (SpecINT x 10^3)
Frequency (MHz)
Typical Power (Watts)
Number of Logical Cores

Motivação

"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, Apple.

Motivação

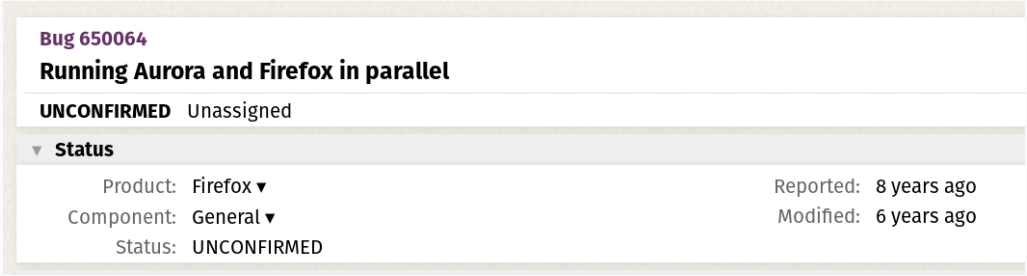
Programação Paralela

Motivação

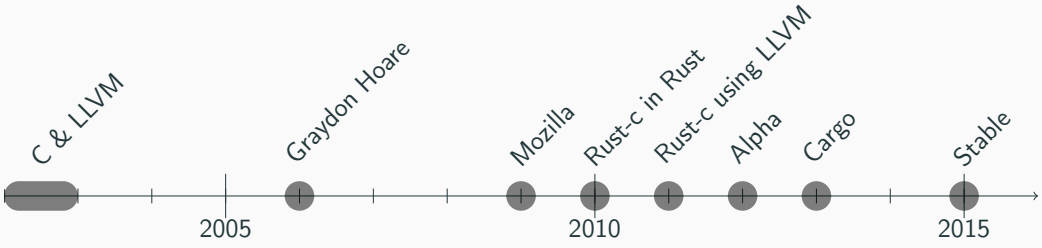
Problemas:

- Data races;
- Deadlock;
- Use After Free;
- Double Free;

Motivação


Bug 650064
Running Aurora and Firefox in parallel
UNCONFIRMED Unassigned
Status
Product: Firefox ▾
Component: General ▾
Status: UNCONFIRMED
Reported: 8 years ago
Modified: 6 years ago

História


C & LLVM
Graydon Hoare
Mozilla
Rust-c in Rust
Rust-c using LLVM
Alpha
Cargo
Stable
2005
2010
2015

Desenvolvimento

- Licença MIT no Github;
- Duas versões: Stable e Nightly;
- Atualizações a cada 6 semanas;
- Processo de RFC;
- Quando uma RFC é aprovada ela é adicionada na versão Nightly;
- Após algum tempo em Nightly, ela pode ser adicionada na versão Stable, deixada de lado ou alterada;

Quem usa? E para que?

Quem usa? E para que?

"Optimizing cloud file-storage."

Quem usa? E para que?

"Everything from server monitoring to middleware!"

Quem usa? E para que?

"Developing memory-safe embedded applications on our SmartThings Hub and supporting services in the cloud."

Quem usa? E para que?

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

Quem usa? E para que?

"Programming Assignments in secured Docker containers."

Quem usa? E para que?

"Letting you develop, deploy and manage infrastructure, run-time environments and applications."

Quem usa? E para que?

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

Quem usa? E para que?

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

Quem usa? E para que?

"Using Rust instead of C/C++ to let you build, deploy and manage packages."

Quem usa? E para que?

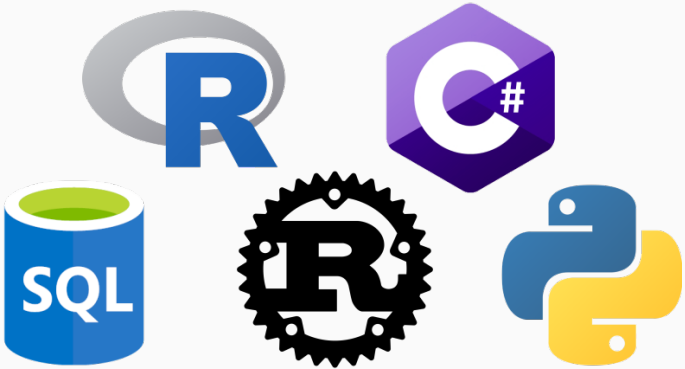
"Extracting anti-fraud insights by analyzing millions of transactions from hundreds of e-commerces."

Quem usa? E para que?

- No site oficial da linguagem há mais 123 empresas que deixaram claro que utilizam Rust em produção;

Sugestões de regras do Velocity

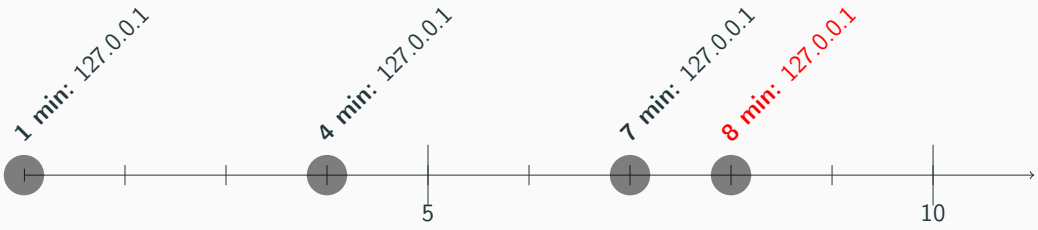
Provas de Conceito



Velocity

Regras de repetição:

- 5 repetições de um **Cpf** em 10 minutos;
- 10 repetições de um **Cartão** em 2 dias;
- 2 repetições de um **Ip** em 5 minutos;



Algoritmo

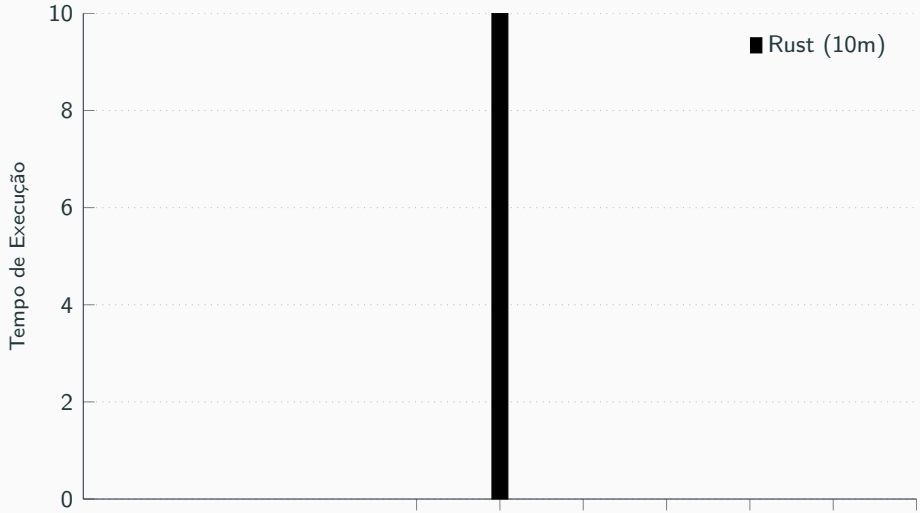
```
var transactions = new List<Transaction>();
var quarantine = new DateTime();

for transaction in transactions do
    if quarantine_is_active(quarantine, transaction) then
        block(transaction);
        update(quarantine);
    else
        if extrapolate_rule(transaction) then
            block(transaction);
            update(quarantine);
        end
    end
end
```

Sobre os Benchmarks V1.0

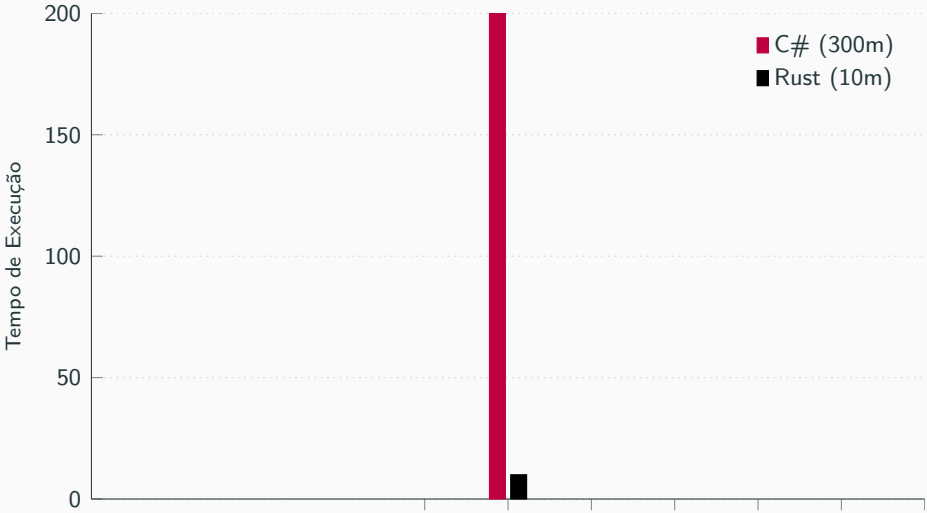
Hardware: I7 8770 (6 núcleos + 6 threads), 8GB RAM DDR4, SSD;

- Todas as versões representam o Velocity de forma exata;
- Não é utilizado agrupamento e nem programação funcional;
- C# e Rust paralelisam o processamento;
- C# e Rust realmente processaram 640 regras, o tempo de R é uma estimativa;
- R executou no banco SQL de monitoria;

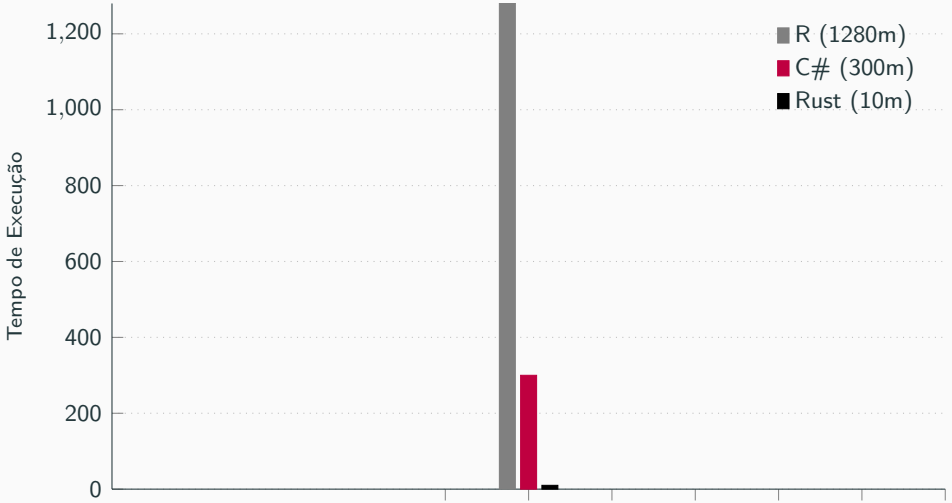
Benchmark (v1.0)



Benchmark (v1.0)



Benchmark (v1.0)

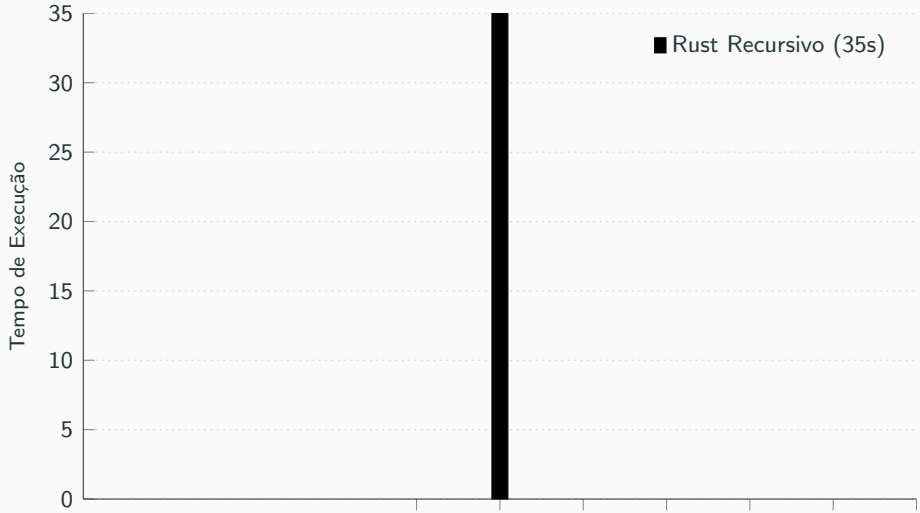


Sobre os Benchmarks V2.0

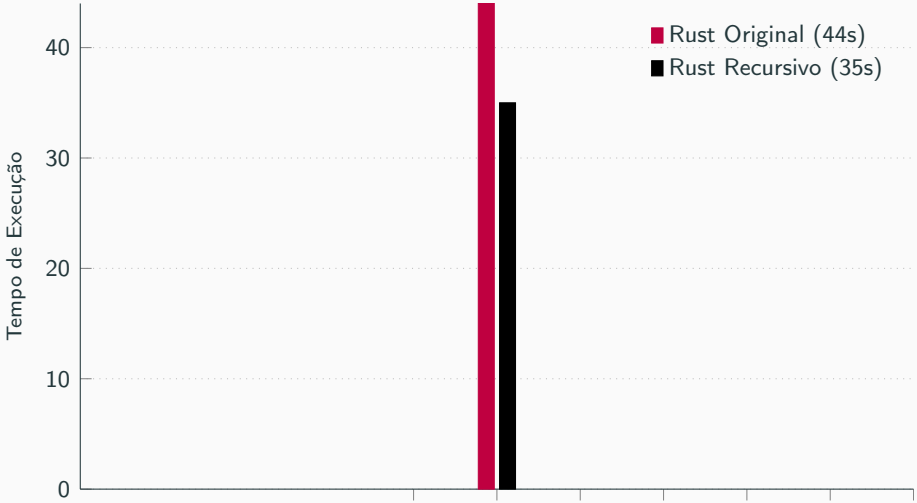
Hardware: I7 8770 (6 núcleos + 6 threads), 8GB RAM DDR4, SSD;

- Todos os algoritmos utilizam agrupamento;
- Rust e Python utilizam programação funcional;
- Duas versões do algoritmo: Original e Recursivo;
- Python utiliza a lib Numpy, que é feita em C, C++ e Fortran;

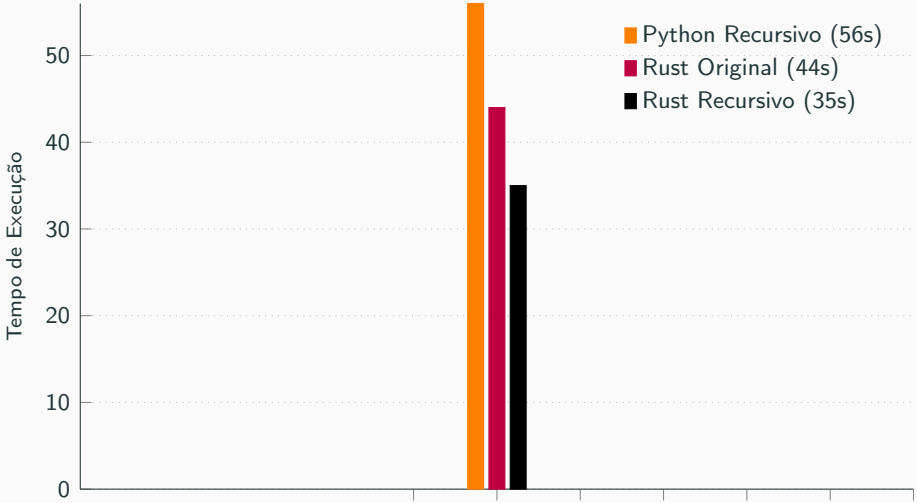
Benchmark (v2.0)



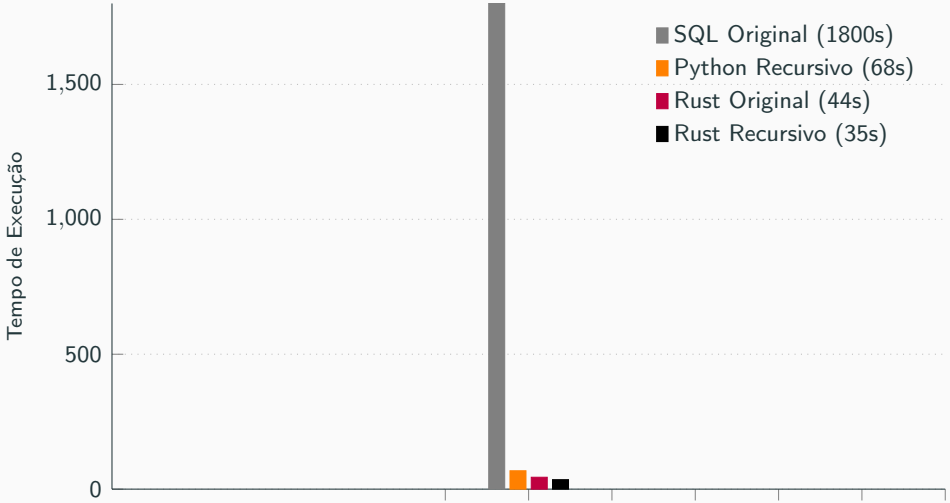
Benchmark (v2.0)



Benchmark (v2.0)



Benchmark (v2.0)



Mostre-me o código

Curiosidades

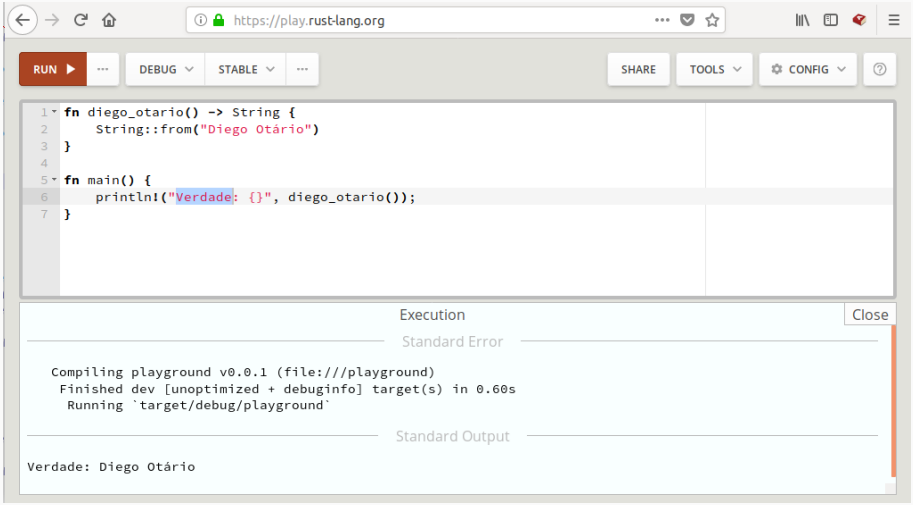
Curiosidades

- Em apenas 3 anos o gerenciador de pacotes de Rust ultrapassou os gerenciadores de R e Haskell em número de pacotes;
- Em 2017 a Red Hat começou a dar suporte para Rust;
- Rust possui uma imagem oficial no Docker Hub;
- Em 2017 todas as IDEs da IntelliJ passaram a dar suporte para Rust;

Curiosidades

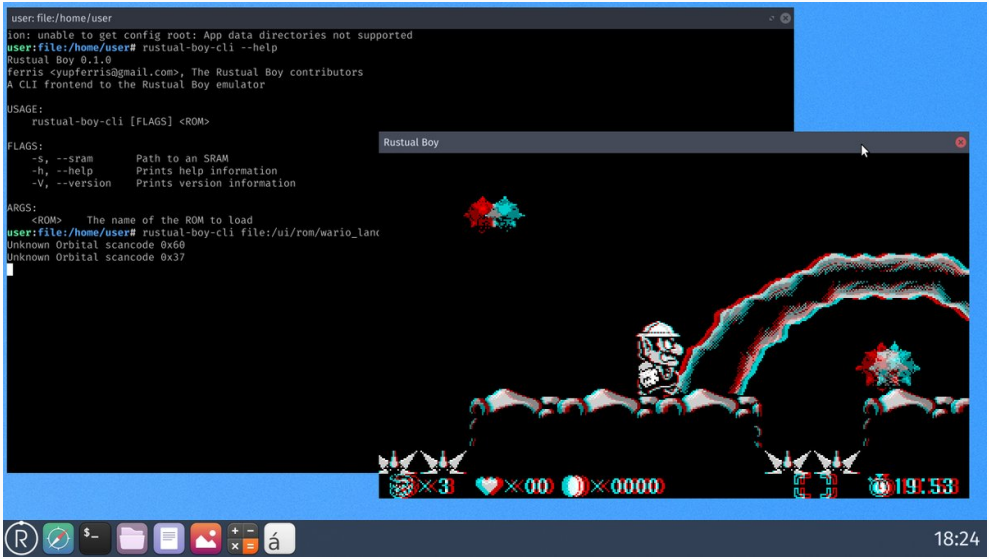
- Amazon e Google estão contratando desenvolvedores Rust;
- Facebook e Github fazem parte dos Gold Sponsors da RustConf;
- Intel encoraja a utilização de Rust;
- Rust possui interoperabilidade com C, C++, C#, Swift, Python e Node.JS;
- Rust possui um gerador automático de APIs para bibliotecas C e C++;

Rust Playground

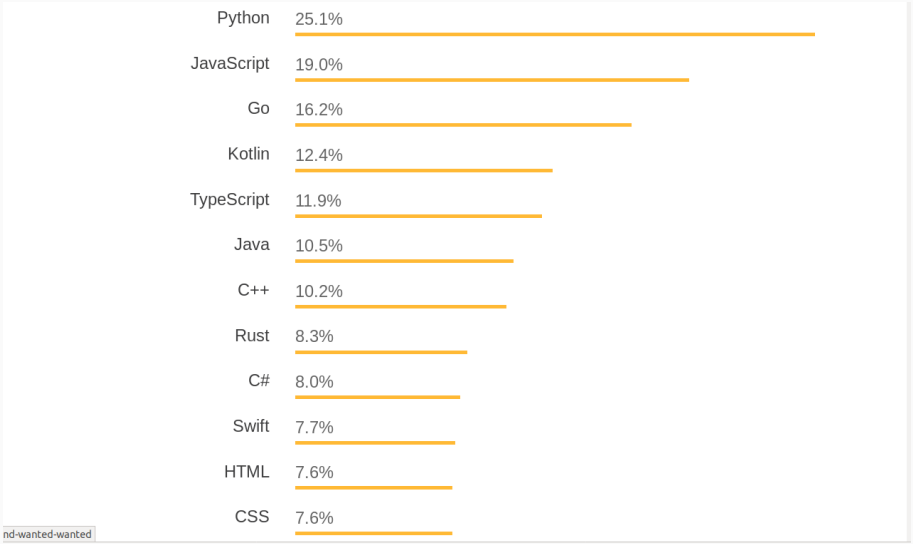


Eventos

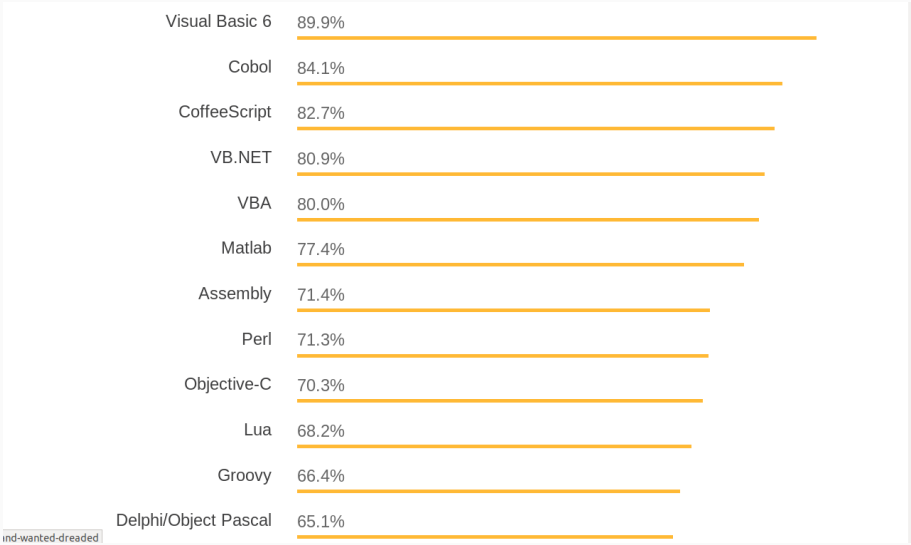
Sistema Operacional



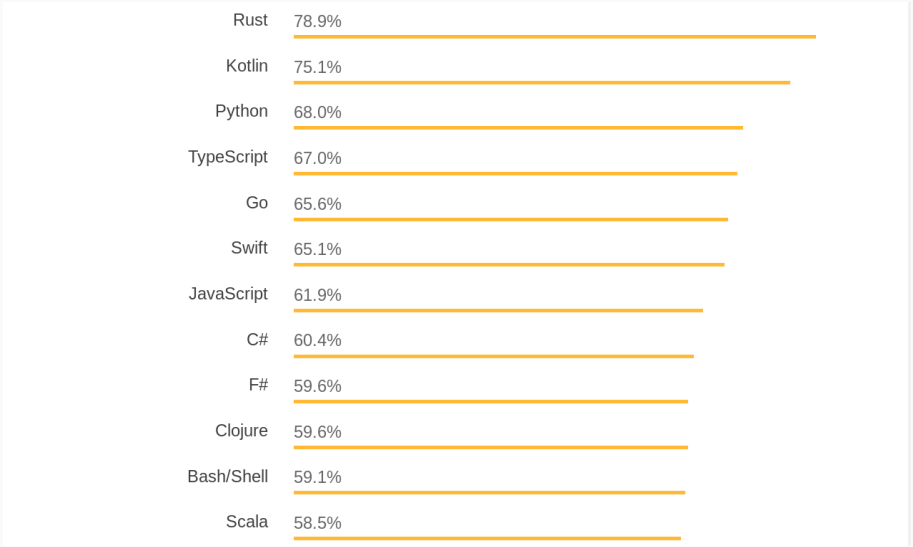
Stackoverflow Insights: Wanted



Stackoverflow Insights: Dreaded

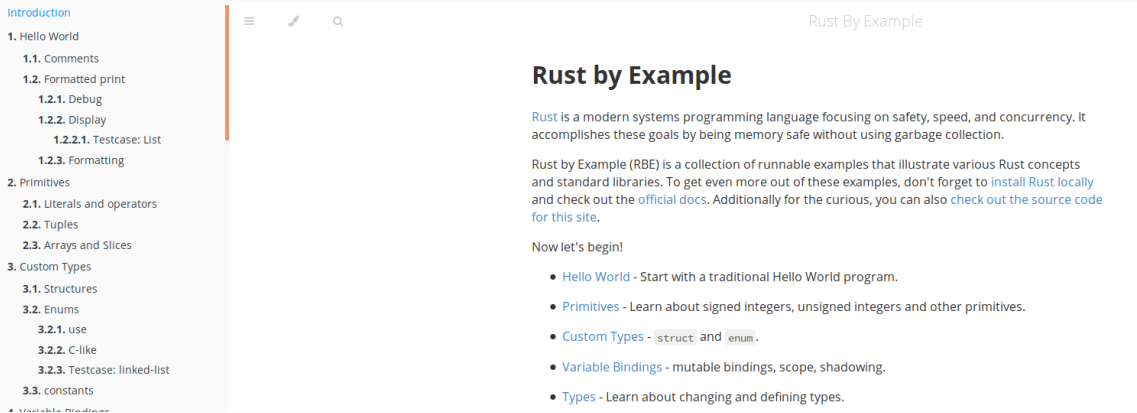


Stackoverflow Insights: Loved



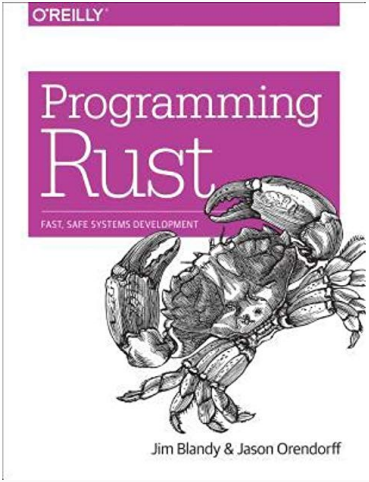
Material de estudos

Rust by Example



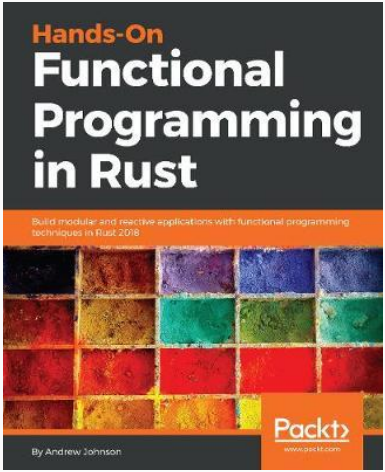

Rust
Concorrência e alta performance com segurança
RUST
Casa do Código
MARCELO CASTELLANI

Programming Rust

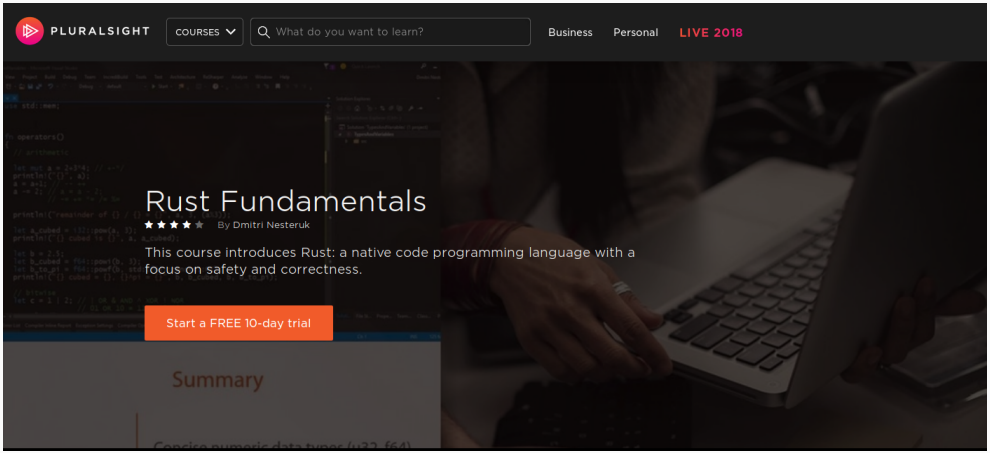


Livro mais popular do Safari Ebooks no mês de seu lançamento em Outubro de 2017

Hands-On Functional Programming in Rust



Rust Fundamentals



Obrigado