

# Eliminando Gargalos de Processamento Utilizando Rust

---

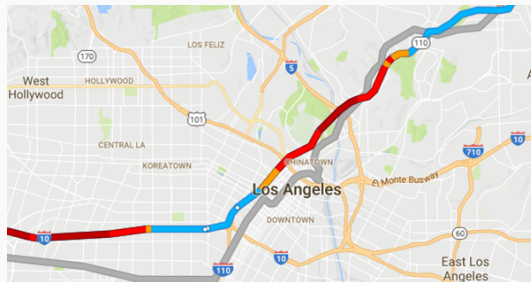
Johnathan Fercher

1. Introdução
2. Quem usa? E para que?
3. Como otimizar um código em Rust
4. Resolução de um problema

# Introdução

---

# Gargalos



- Um gargalo é a parte mais lenta de um sistema:
  - 90% de um trajeto é feito a 110 km/h e 10% é feito a 20 km/h;
  - 100 pessoas precisam de atendimento médico por dia, porém apenas 10 são atendidas;

- IO-Bound;
- CPU-Bound;

- IO-Bound → Async Programming;
- CPU-Bound → Algorithms, Parallel Programming, Programming Languages;



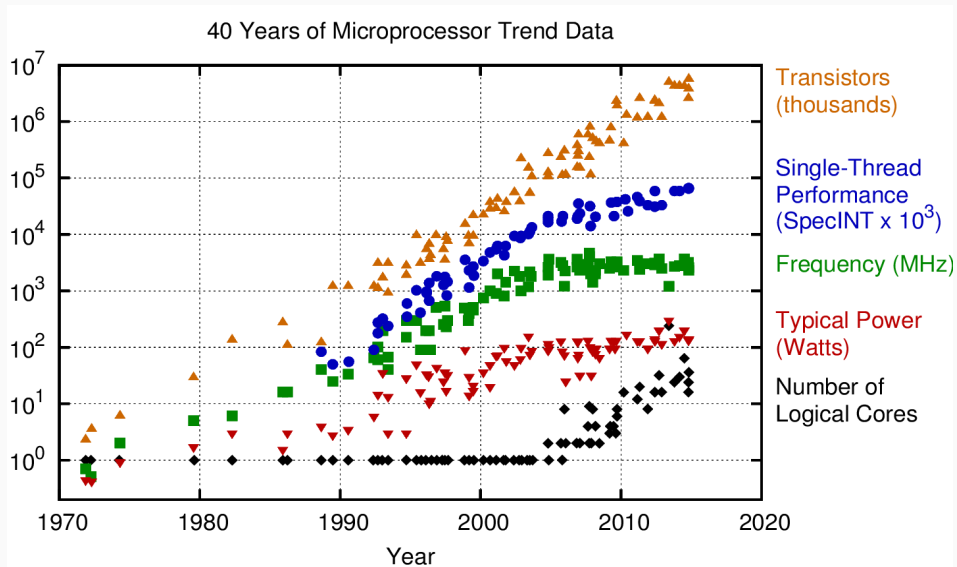
- Programming Language;
- Memory and thread-safe;
- Rust  $\rightarrow$  LLVM  $\rightarrow$  EXE;
- C-Bindings;
- Object-Oriented and Functional;
- Unit-tests and Package Manager;
- Interfaces and Generics;
- Without Garbage Collector;

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

*Lei de Moore, 1965.*



# 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.*

**Bug 650064**

## Running Aurora and Firefox in parallel

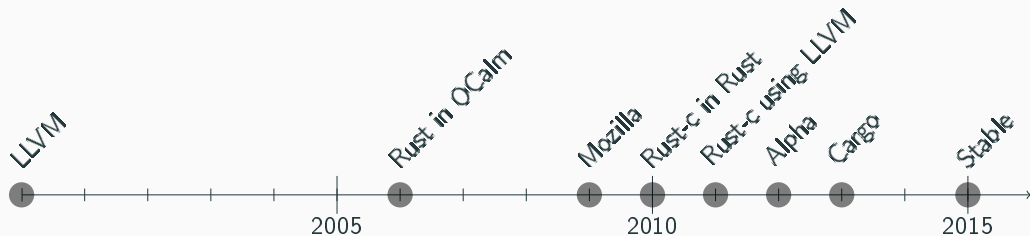
**UNCONFIRMED** Unassigned

### ▼ Status

Product: Firefox ▼  
Component: General ▼  
Status: UNCONFIRMED

Reported: 8 years ago  
Modified: 6 years ago

# História



- 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."

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

"Everything from server monitoring to middleware!"





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

**moz://a**

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



"Programming Assignments in secured Docker containers."



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



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

## Quem usa? E para que?

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

# Como otimizar um código em Rust

---

## Exemplos

```
let mut primes = Vec::new();

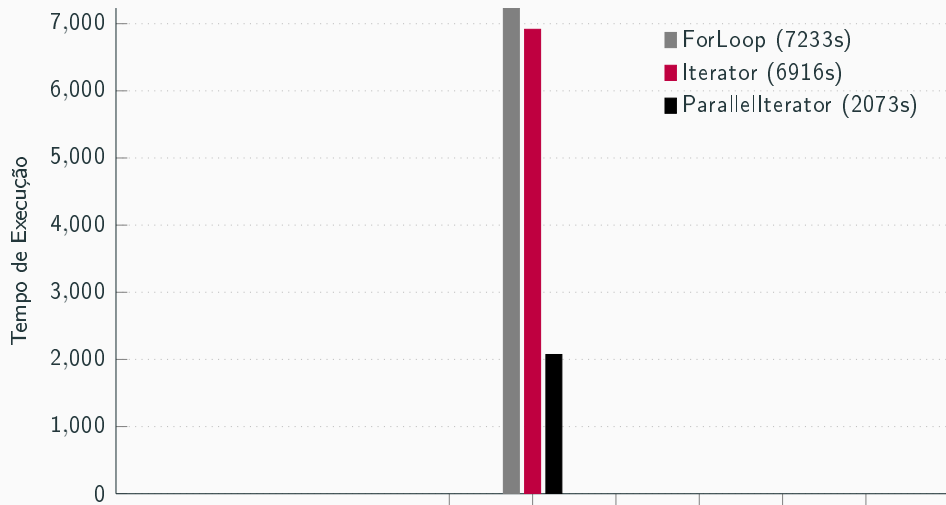
for number in numbers {
    if is_prime(number) {
        primes.push(number);
    }
}
```

```
let primes = numbers.iter()
    .filter(|x| is_prime(x))
    .collect::<Vec<u64>>();
```

```
let primes = numbers.par_iter()
    .filter(|x| is_prime(x))
    .collect::<Vec<u64>>();
```



## Benchmark para 150000 números

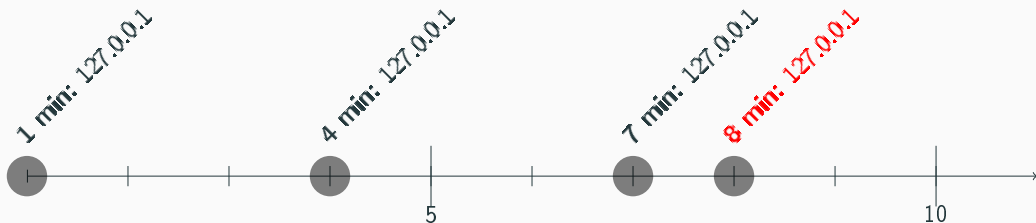


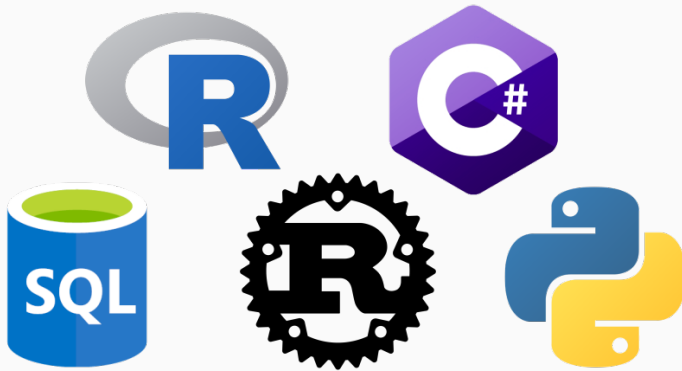
## Resolução de um problema

---

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;

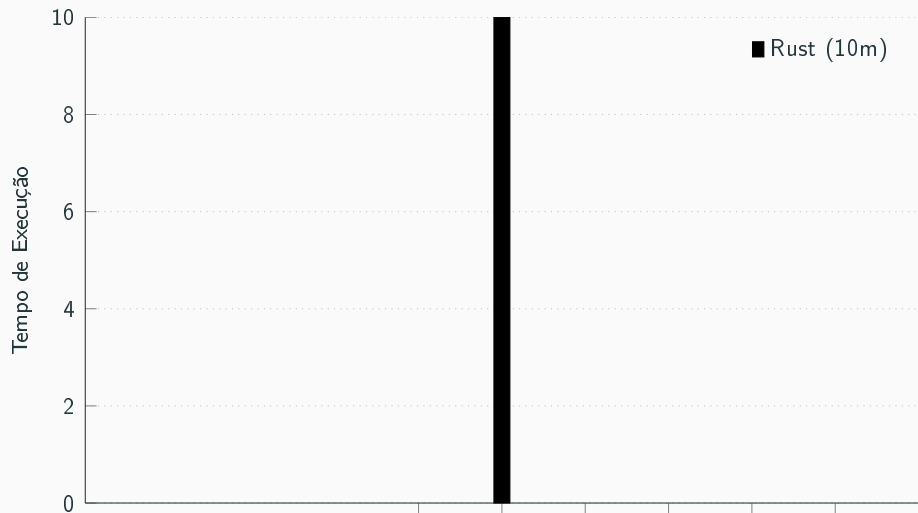




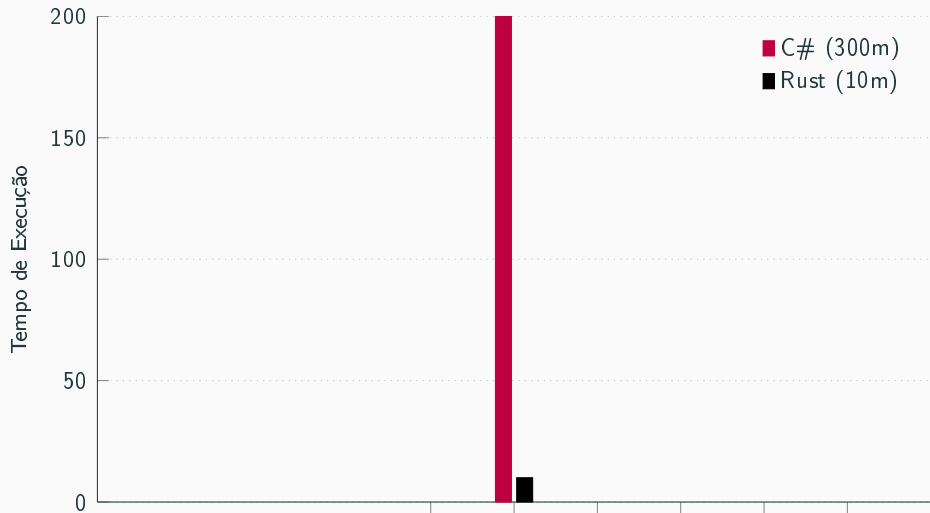
**Hardware: I7 8770 (6 núcleos + 6 threads), 8GB RAM DDR4, SSD;**

- Não é utilizado nenhuma técnica de agrupamento;
- Não é utilizado nenhum artifício de programação funcional;

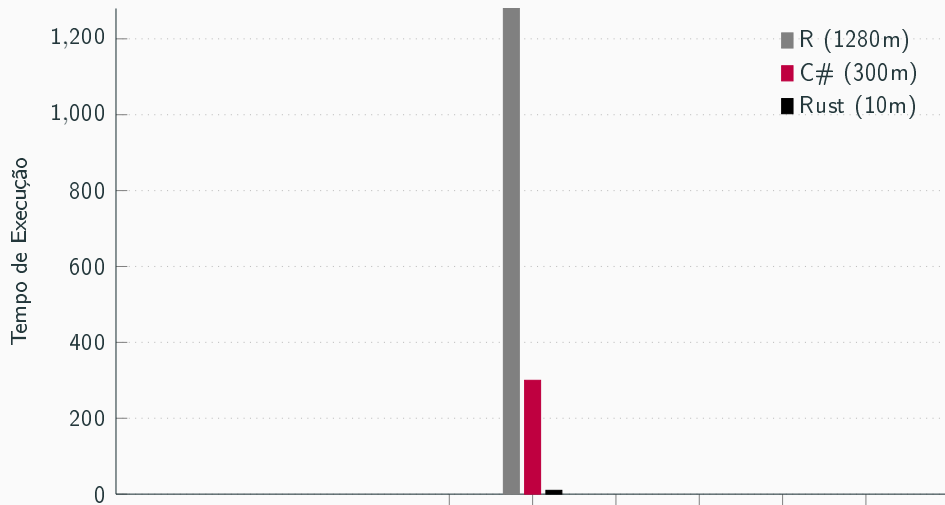
## Benchmark (v1.0)



# Benchmark (v1.0)



# Benchmark (v1.0)

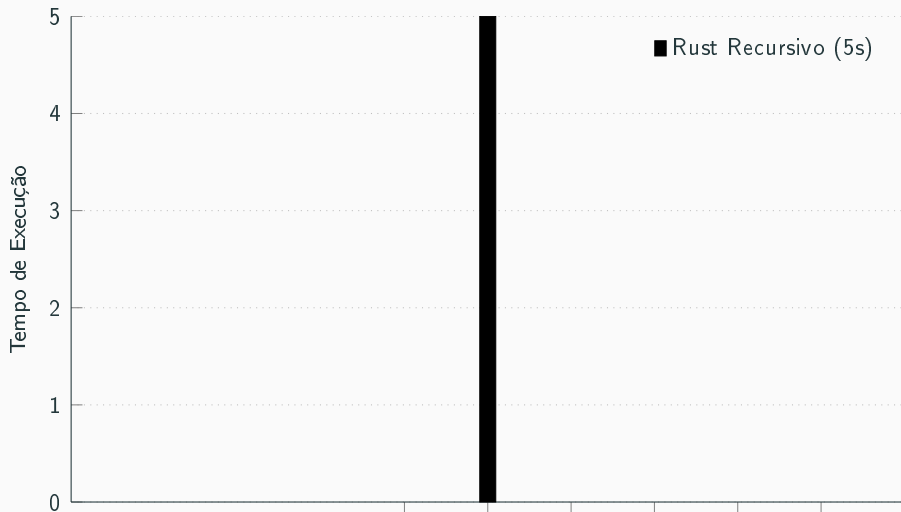




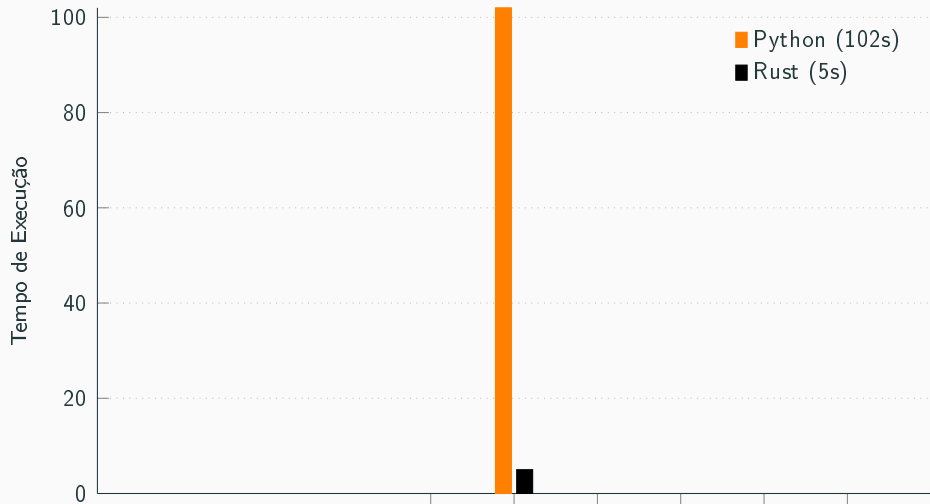
**Hardware:** I7 8770 (6 núcleos + 6 threads), 8GB RAM DDR4, SSD;

- Todos os algoritmos utilizam agrupamento;
- Rust e Python utilizam programação funcional;
- Python utiliza a lib Numpy, que é feita em C, C++ e Fortran;

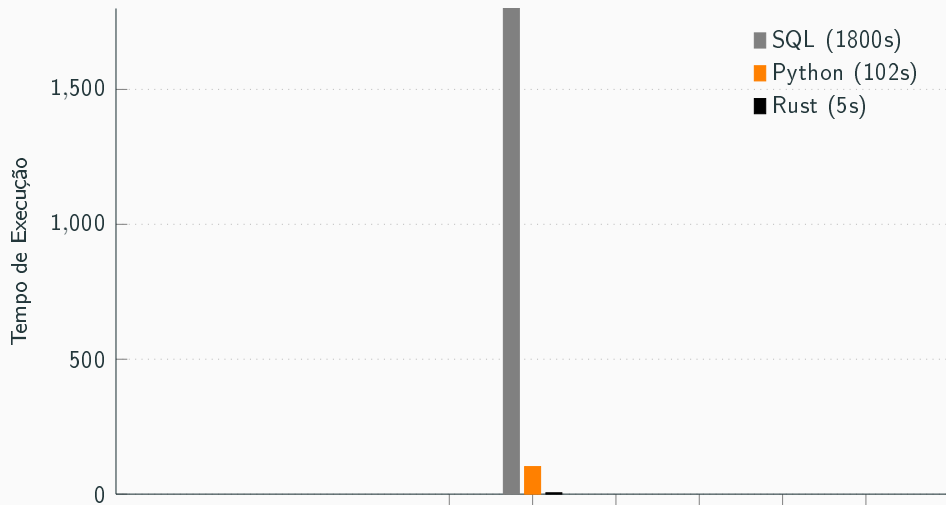
# Benchmark (v2.0)



## Benchmark (v2.0)



# Benchmark (v2.0)





Obrigado