

---

# Escrevendo scripts cross-plataformas com Rust

Fácil, rápido e portátil!

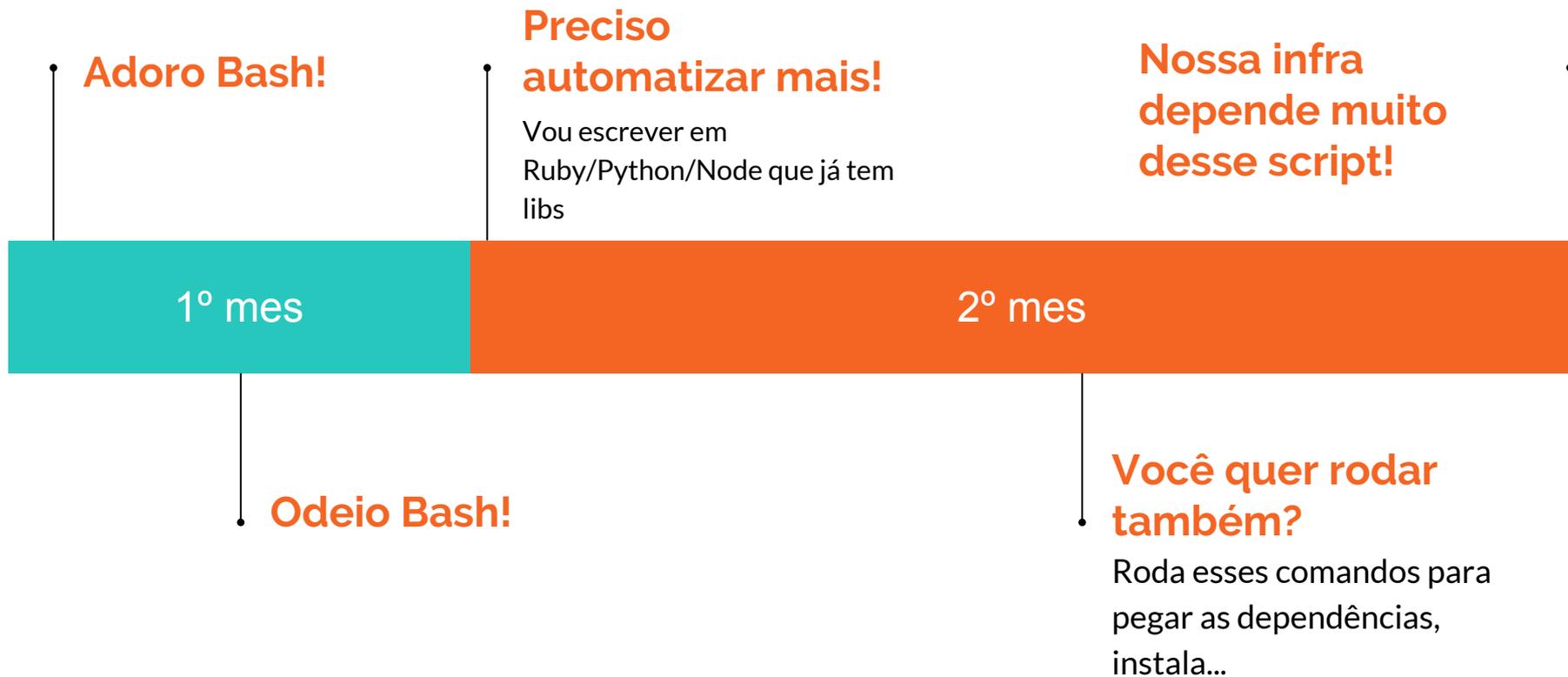
---



—  
**Preciso automatizar esse  
processo...**

**Já sei, vou fazer um script!**

# O ciclo de vida daquele pequeno script



# E se fosse possível escrever um script:

- Que não precisa de dependências para rodar
- “Copia o arquivo e roda”
- E que pode ter testes

(E sem Docker se você quiser!)



# Vamos usar Rust!

É só seguir esses passos para começar o seu projeto:

<https://rustup.rs/>

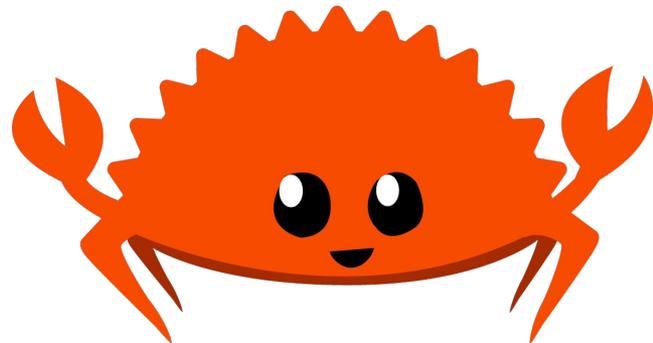
rustup is an installer for  
the systems programming language [Rust](#)

Run the following in your terminal, then follow  
the onscreen instructions.

```
curl https://sh.rustup.rs -sSf | sh
```

Need help? [Ask on #rust-beginners](#).

 rustup is an official Rust project.  
[other installation options](#) - [about rustup](#)



## 1. Criar um projeto

```
$ cargo new import-ids
```

## 2. Adicionar uma dependência

```
# Cargo.toml  
  
[dependencies]  
  
clap = "2"
```

## 3. Rodar

```
$ cargo run
```

```
$ cargo run
```

```
    Compiling import-ids v0.1.0  
(file:///home/bruno/import-ids)  
    Finished dev [unoptimized + debuginfo] target(s)  
in 0.56 secs  
    Running `target/debug/import-ids`  
Hello, world!
```



```
# src/main.rs  
  
fn main() {  
    println!("Hello, world!");  
}
```

```
# src/main.rs

extern crate clap; // Importamos nossa dependencia

use clap::{App, SubCommand}; // Avisamos que vamos ter um App e Subcomandos

fn main() {

    let args = App::new("import-ids")

        .version("1.0")

        .about("Does awesome things")

        .subcommand(SubCommand::with_name("import")

            .about("does importing things"))

        .get_matches();

    if let Some(_import_args) = args.subcommand_matches("import") {

        println!("importing things");

    }

}
```

```
$ ./target/debug/import-ids import
```

```
importing things
```

# Pronto!

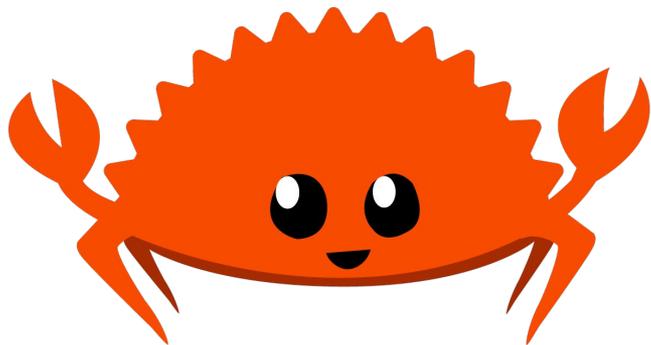
Debug? Quero otimizar o binário!

```
$ cargo build --release
```

Agora só precisamos copiar!

```
aws s3 cp \  
target/release/import-ids.exe \  
s3://clis/windows/import-ids.exe
```

Perai?! Windows?



## Cross compilação

[jparic/cross](https://github.com/jparic/cross) - Para gerar binarios para outras plataformas

[jparic/trust](https://github.com/jparic/trust) - Para criar pipelines de build multiplataformas no Travis/AppVeyor

# Conheça a comunidade

Existe um grupo oficial focado em melhorar a experiência para criar apps para linha de comando

- [CLI Work Group](#)

## IRC Oficial

[#rust](#), [#rust-beginners](#) e [#rust-br](#)

Comunidade brasileira no Telegram  
[@rustlangbr](#)

---

# Existe uma certa curva de aprendizagem...

Mas o pessoal está bem disposto a ajudar.

A comunidade se preocupa muito com quem está iniciando os estudos.

# Mas os programas ficam bem rápidos!

## **Versão ingênua com grep / uniq / sort / ...**

1000 linhas - 1-2 segundos

20.000 linhas - 5 minutos

160.000 linhas - 15 minutos

## **Versão Rust**

4.000.000 linhas - 14 segundos



## “Sei não... Prefiro meus scripts ainda”

Existem vários apps super rápidos que são escritos em Rust e que você pode começar a usar hoje.

Seus scripts ficarão mais rápidos, e você não precisa instalar nada de Rust.

# Conheça:

## ripgrep

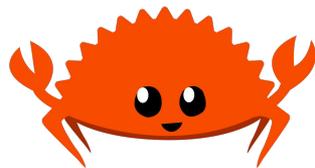
Um *grep* muito rápido!

## xsv

Um *jq* para lidar com arquivos em CSV

## fd

Uma alternativa ao *find* rápido e mais amigável



Se tiver quaisquer dúvidas em como começar, sou o **Bruno Tavares** ([@bltavares](#)) na internet e podem fazer perguntas.

Me manda um email que te dou dicas dependendo do que você já conhece!