

Parte 2

# **Rust na vida real**

## **Por que?**

- Vocês já sabem: eficiência e confiança.

## **Por que não?**

- Vocês também já sabem: tecnologia muito nova, um tanto complicada, e que não se adapta a todos os modelos de negócio.

## **Por que não... ainda?**

- É uma pergunta mais interessante.

# Por que não... ainda?

- **Sistemas muito críticos (bancários, médicos, etc.):** já existem tecnologias legacy muito bem estabelecidas nos usuários deste tipo de sistema. Reescrevê-las usando uma linguagem com pouco mais de 7 anos de estabilidade não é uma boa ideia. Isso pode mudar com o amadurecimento da linguagem.
- **Computação científica:** muitas vezes a segurança não é crítica, e reescrever um ambiente já estabelecido (em Python e/ C++, por exemplo), mais atrasaria a pesquisa do que ajudaria. Isso já está mudando.
  - [Nature. Why scientists are turning to Rust.](#)

# Por que não... ainda?

- Areweyet?

Name	Url	Description	Owner
Are we async yet?	<a href="https://areweasynyet.rs/">https://areweasynyet.rs/</a>	Asynchronous I/O in Rust	Web Services WG
Are we audio yet?	<a href="https://areweaudioyet.com/">https://areweaudioyet.com/</a>	Audio related development in Rust	RustAudio
Are we Chrome yet?	<a href="http://arewechromeyet.com/">http://arewechromeyet.com/</a>	Parity-Chrome bugs	??
Are we compressed yet?	<a href="https://arewecompressedyet.com/">https://arewecompressedyet.com/</a>	Video codec quality	Xiph.Org Foundation
Are we distributed yet?	<a href="https://arewedistributedyet.com/">https://arewedistributedyet.com/</a>	Peer-to-peer web experience	<a href="https://github.com/arewedistributedyet/arewedistributedyet">https://github.com/arewedistributedyet/arewedistributedyet</a>
Are we everyone yet?	<a href="http://areweeveryoneyet.org/">http://areweeveryoneyet.org/</a>	Total, Employees, and Volunteer contributors per Firefox release.	Mike Hoyer
Are we fast yet?	<a href="http://arewefastyet.com/">http://arewefastyet.com/</a>	Tracking performance of popular JavaScript engines	JavaScript team
Are we fun yet?	<a href="http://arewefunyet.com/">http://arewefunyet.com/</a>	Web platform game development	Martin Best
Are we game yet?	<a href="http://arewegameyet.com/">http://arewegameyet.com/</a>	Rust game development	doppioslash
Are we GUI yet?	<a href="http://areweguiyet.com/">http://areweguiyet.com/</a>	Rust GUI development	<a href="https://github.com/areweguiyet/areweguiyet">https://github.com/areweguiyet/areweguiyet</a>

# ***Systems programming***

- Programação de softwares de sistema: softwares feitos como plataforma para outros softwares.
- Vão desde sistemas operacionais, a *engines* de videogames, a aplicações em **software como serviço (SaaS)**.
- Usualmente é feita em linguagens compiladas, muito eficientes, e que de preferência permitam que o programador escreva trechos do código em linguagem de montagem.
- Rust foi construída com esse objetivo.

Quem? 



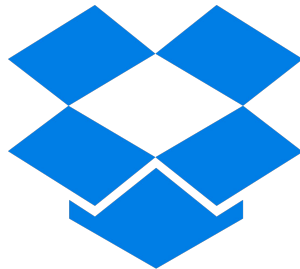
stone



magnetis

Referência: <https://github.com/rust-br/eu-uso-rust>

**Quem?**



# Referências (empresas)

## Meta

1. [A brief history of Rust at Facebook](#)
2. [Programming languages endorsed for server-side use at Meta](#)

## Amazon

1. [Why AWS loves Rust, and how we'd like to help](#)

## Microsoft

1. <https://github.com/krustlet/krustlet>
2. [In Rust We Trust: Microsoft Azure CTO shuns C and C++](#)

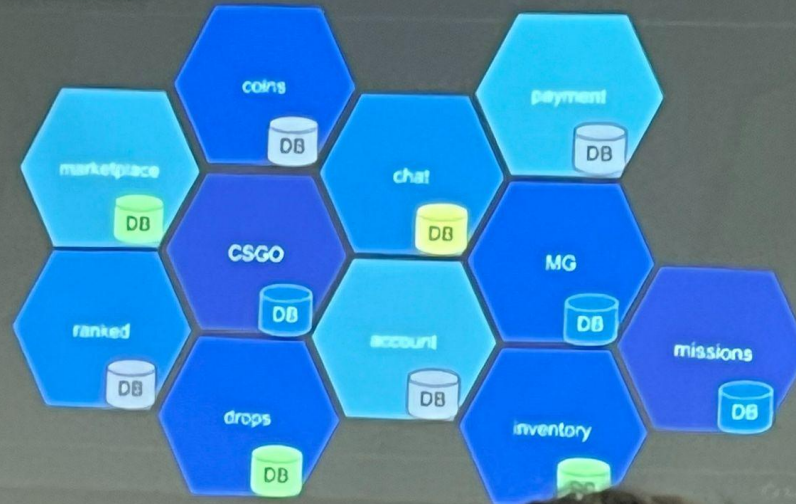
(entre outros projetos internos, em especial de laboratórios – alguns artigos dão a entender que eles também usam internamente no Azure)

Restante: [Rust. Usuários em produção.](#)



# Onde?

## Contexto





S | BATE-PAPO | ENTREVISTA  
DESAFIOS | ATIVIDADES PRÁTICAS  
17 a  
anos de ciência, tecnologia e in

# Onde? Microserviços




# Onde? Microsserviços

 rust-lang.org/production/users




## SmartThings

Memory-safe embedded applications on our [SmartThings Hub](#) and supporting **services** in the cloud.



## Bitfury Group

[Exonum is an extensible framework for blockchain projects written in Rust.](#)



## Kodebox

We created CodeChain, a programmable open source blockchain technology optimal for developing and customizing multi-asset

service| 9/24 ^ v x

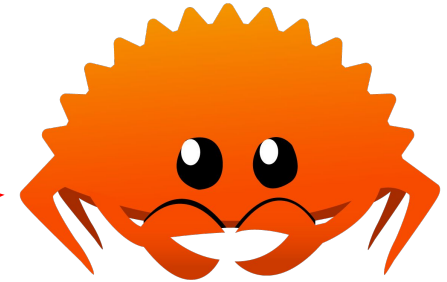
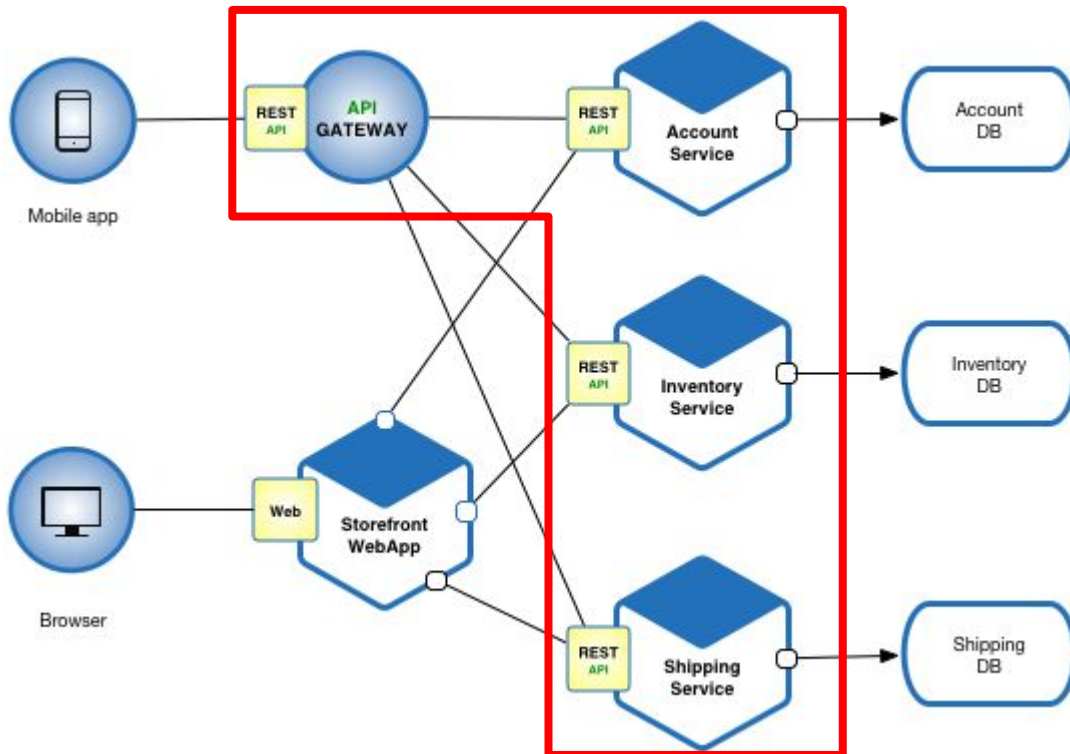
# Onde? Microserviços

- Discord: serviço de Read States.
- TerraMagna: “trabalhando com Rust em todos os microserviços”.
- iFood: Usamos Rust em uma série de microserviços dentro da nossa plataforma logística.
- Facebook: servidores e serviços internos
- Amazon: serviços da AWS
- Microsoft: escalabilidade, serviços do Azure



# Microserviços

- Essencialmente dividir para conquistar.
- Nessa divisão, surgem muitos sistemas...
- ...aí o Rust brilha.



# Onde mais?

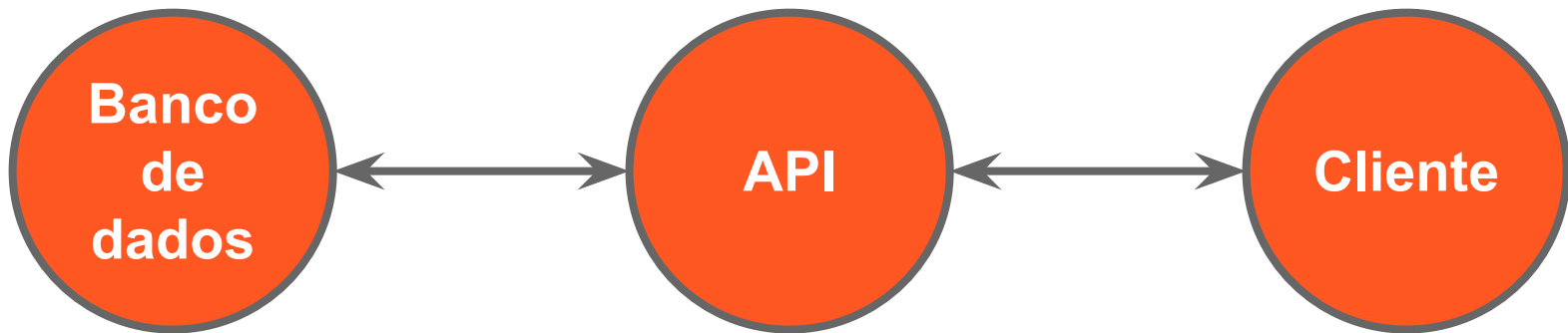
- Não somente em SaaS ou em empresas que fornecem serviços web, mas qualquer coisa que se caracterize como um sistema, como:
  - Serviços de sistemas operacionais (daemons)
  - Software de servidor
  - Processamento interno de dados
- Muito usada hoje em dia pois, com a abordagem de microsserviços, muitas coisas que não eram sistemas hoje tornaram-se sistemas.
- Basicamente substitui C e C++ em vários usos. Mesmo com suas complicações, compensa pela eficiência, segurança e, principalmente, pelo ecossistema: com cada vez mais bibliotecas “prontas” surgindo, a tendência é que o uso da linguagem aumente ainda mais, pois isso elimina em boa parte a dificuldade de usá-la, que seria criar algo do zero, respeitando todas as suas regras.

Sem mais delongas...

**Let's dance!**

# Construindo nosso “microserviço”

- Bem simples, apenas um recorte.





# Vamos fazer uma discoteca!

- Sentido literal da palavra: uma biblioteca de álbuns musicais.
- Antes de começar, vamos precisar de alguns conceitos:
  - API
  - API REST
  - HTTP Requests
  - SQL
  - Chave-valor
  - ORM

# O que vamos usar? (além do Rust)



Hoppscotch



Allow CORS: Access-Control-Allow-Origin

# O que vamos usar? (no Rust)



ORM para Rust

+



HTTP para Rust

# O que vamos usar? (no Rust)



**DIESEL**

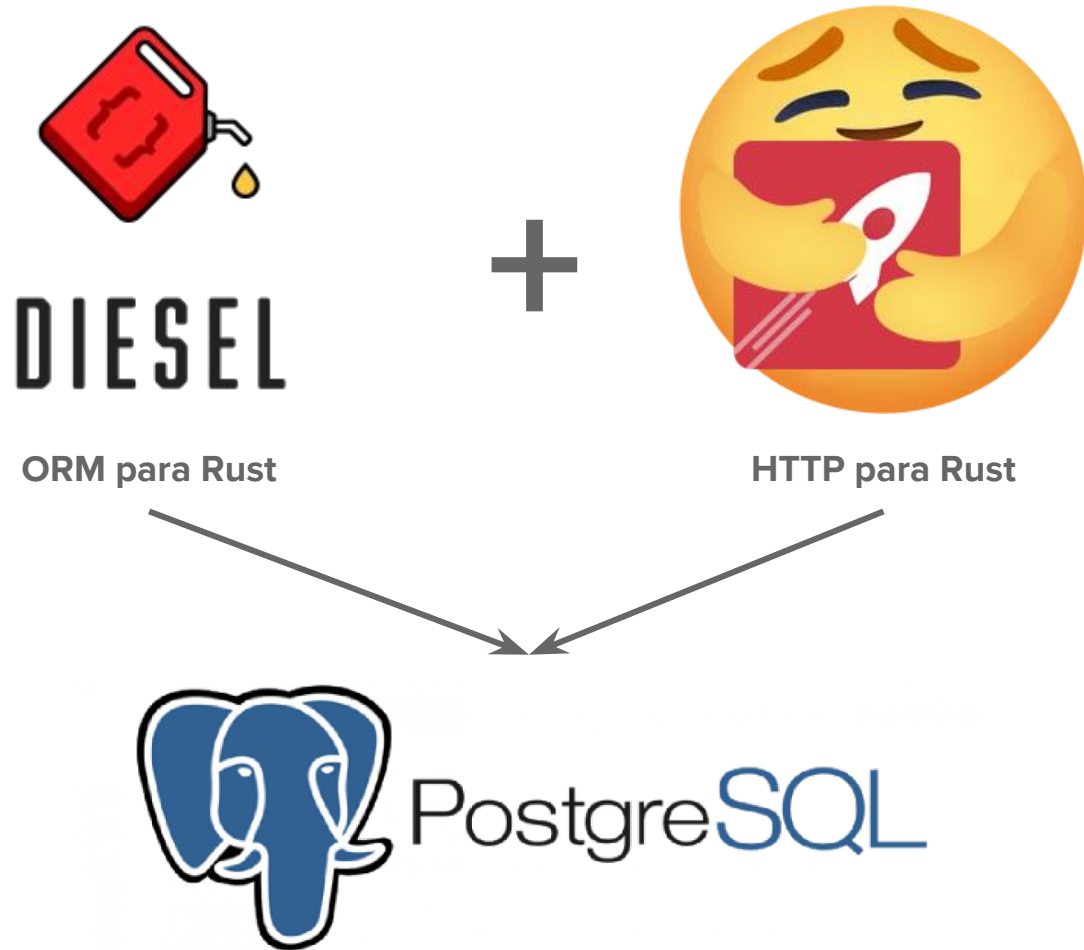
ORM para Rust

+



HTTP para Rust

# O que vamos usar? (no Rust)



# Modelo do banco de dados

albums	
PK	<u>id serial</u>
	artist varchar(64) not null
	title varchar(64) not null
	cover_art_url varchar(256) not null
	label varchar(32) not null
	release_date date not null

# Nosso microserviço



~~NEXT~~.JS

# Tarefa de casa!



**Tarefa (calma, é só uma brincadeira pra quem gosta de web):** Substituir o Next.js pelo fresh, um framework web construído em Deno.

Deno, caso lembrem dos primeiros slides, é um runtime para JavaScript escrito em Rust! Ou seja, podemos usar o Rust no nosso front também.

No exemplo do minicurso, utilizamos o Next.js para tentar simular um caso mais “real”, pois o Deno ainda é pouco usado em produção.





**That's all Folks!**  
**Obrigado.**