



# **Rust para potenciar la Inteligencia Artificial**



# **Rust en Español**

# Sergio Ribera

 [bento.me/sergioribera](https://bento.me/sergioribera)

 [@SergioRibera](https://twitter.com/SergioRibera)

 [/in/sergioribera](https://www.linkedin.com/in/sergioribera)

# Relevancia de la IA

Como tecnología emergente



**LLaMA**  
by  Meta



# **Inversion en los ultimos años**

**93.5 B \$   121.5 B \$   158.0 B \$**

# Retos en la IA



Tiempo



Rendimiento



Costos

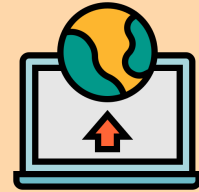
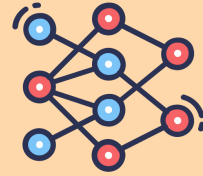


Portabilidad



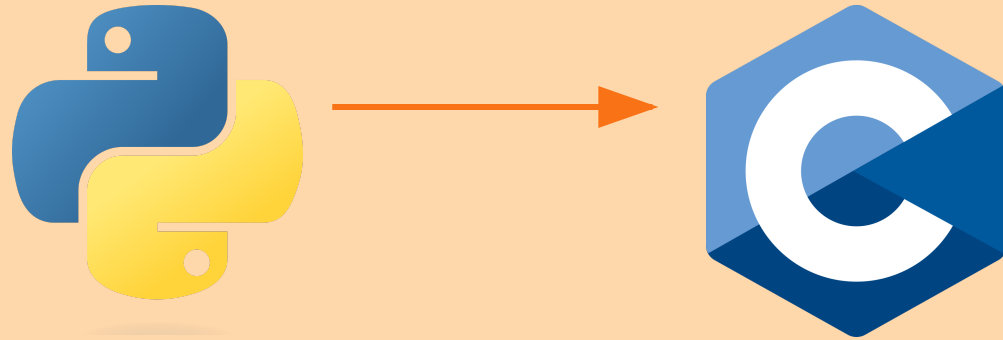
Huella de  
Carbono

# ¿Como funciona la IA?





# Librerías Actuales



# El Problema

```
#include <stdio.h>

int main() {
    int *ptr = NULL;

    // Intento de escribir en un puntero nulo
    *ptr = 5;

    return 0;
}
```



```
→ gcc main.c -o main && ./main
Segmentation fault (core dumped)
```

# Librerías para IA

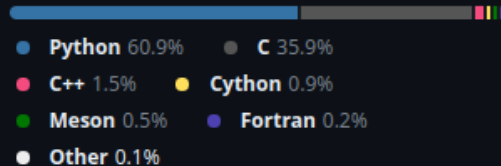
## Pandas

### Languages



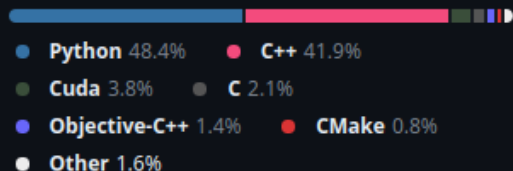
## NumPy

### Languages



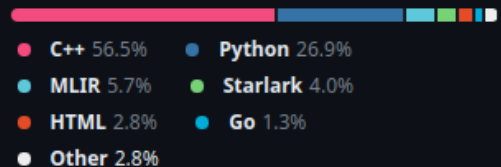
## PyTorch

### Languages



## TensorFlow

### Languages



# ¿Porque Rust?

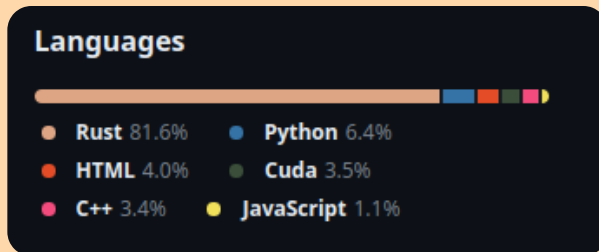
- Multiparadigma
- Herramientas modernas
- Memory safe (null safety, sin fugas de memoria, etc)
- Concurrente de nacimiento
- Sin Garbage Collector
- Abstracciones sin costo
- Enfocado en el rendimiento
- Compilador inteligente

# Librerías para IA con Rust

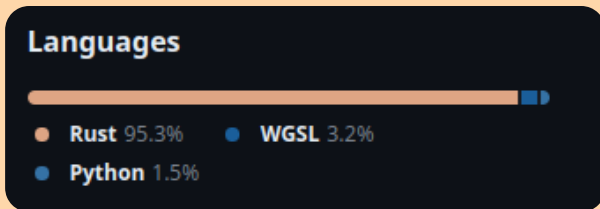
## Polars



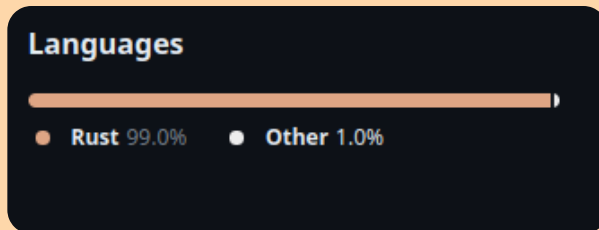
## candle



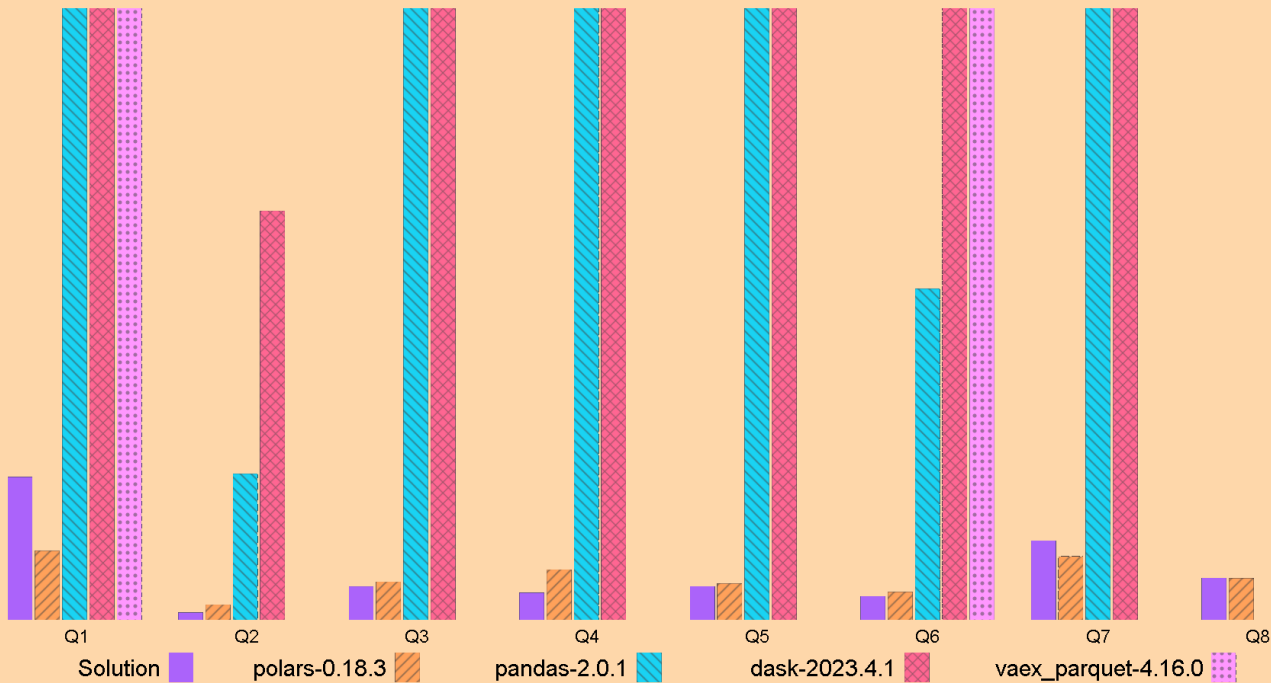
## burn



## leaf






# Librerías para IA con Rust



**Py03**

# Pero, ¿Ya se usa en Inteligencia Artificial?

 **xAI**   
@xai · Follow






Announcing Grok!

Grok is an AI modeled after the Hitchhiker's Guide to the Galaxy, so intended to answer almost anything and, far harder, even suggest what questions to ask!

Grok is designed to answer questions with a bit of wit and has a rebellious streak, so please don't use... [Show more](#)

4:51 AM · Nov 5, 2023

 45.8K  Reply  Copy link

Read 5.4K replies

identified and automatically handled. At xAI, we have made maximizing useful compute per watt the key focus of our efforts. Over the past few months, our infrastructure has enabled us to minimize downtime and maintain a high Model Flop Utilization (MFU) even in the presence of unreliable hardware.

Rust has proven to be an ideal choice for building scalable, reliable, and maintainable infrastructure. It offers high performance, a rich ecosystem, and prevents the majority of bugs one would typically find in a distributed system. Given our small team size, infrastructure reliability is crucial, otherwise, maintenance starves innovation. Rust provides us with confidence that any code modification or refactor is likely to produce working programs that will run for months with minimal supervision.

We are now preparing for our next jump in model capabilities, which will require reliably coordinating training runs on tens of thousands of accelerators, running

	Energy
(c) C	1.00
(c) Rust	1.03
(c) C++	1.34
(c) Ada	1.70
(v) Java	1.98
(c) Pascal	2.14
(c) Chapel	2.18
(v) Lisp	2.27
(c) Ocaml	2.40

	Time
(c) C	1.00
(c) Rust	1.04
(c) C++	1.56
(c) Ada	1.85
(v) Java	1.89
(c) Chapel	2.14
(c) Go	2.83
(c) Pascal	3.02
(c) Ocaml	3.09

# Empresas que usan Rust





# Empresas que usan Rust







# **Rust en Español**





# RustLangES

 [rustlanges.github.io](https://rustlanges.github.io)

 [rustlanges.github.io/blog](https://rustlanges.github.io/blog)

 @RustLangES

 [discord.gg/4ng5HgmaMg](https://discord.gg/4ng5HgmaMg)

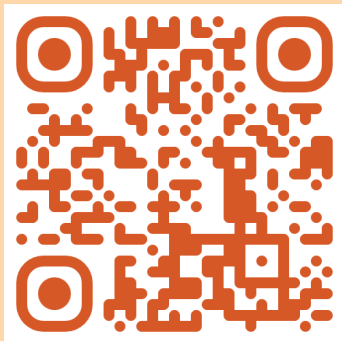
 [/company/rustlanges](https://www.linkedin.com/company/rustlanges)



**Web**



**Blog**



**Github**



**LinkedIn**



# **Rust en Español**