

# I.A. Full Cycle com TypeScript

Isso mesmo: tipos, *console.log()* e código com  
chaves!

TDC POA – 2024

I.A. Full Cycle com TypeScript

**Quem sou eu**

**Laion Azeredo**

webdev 12 yrs+

lead dev @ ADP

**!(Especialista em I.A.)**

Laion Azeredo - 2024



## Agenda

- Porque Python domina o campo de I.A.
- JS/TS e I.A.: O Ecossistema atual
- *Talk is cheap, show me the code!*
  - *Data wrangling* com Jupyter, Deno e Danfo.js
  - Machine Learning com TensorFlow.js

## Uma breve história de Python ❤️ IA (1)

- Python surge em 1991 como um 'projeto de natal' de Guido Van Rossum;
- A ideia era criar uma linguagem 'tão fácil de ler quanto o inglês' e que fosse de uso prático para tarefas simples do dia a dia;

## Uma breve história de Python ❤️ IA (2)

- Principalmente após a publicação do Python 2.0, a linguagem começa a se tornar bastante popular entre a comunidade Linux, em especial para a programação de utilitários;
- Mas a *virada de chave* para a adoção do Python no ramo científico foi o lançamento do **Numpy** em 2005;

## Uma breve história de Python ❤️ IA (3)

- O Numpy traz para o jogo estruturas de dados novas para lidar de modo eficiente com arrays multidimensionais e operações matemáticas complexas;
- Com um *backend* em **C** (para a maioria das operações com arrays) e **Fortran** (para a parte de Álgebra Linear, em especial), o desempenho do Numpy, associado com a simplicidade do Python, fizeram o ecossistema crescer exponencialmente na comunidade científica;

## Uma breve história de Python ❤️ IA (4)

Na esteira do Numpy e do crescimento do Python para fora da comunidade *hacker* do Linux, vieram ao mundo diversas bibliotecas que compõem a base do ecossistema e sua força atual;

## Uma breve história de Python ❤️ IA (5)

- Ainda em 2005, o **SciPy** consolida sua infraestrutura em torno do Numpy;
- Em 2007 temos o lançamento do **scikit-learn**, trazendo diversos algoritmos de *machine learning* com uma interface amigável e eficiente;
- Em 2008 chega o **Pandas** e seu *Dataframe* para lidar com dados tabulares;
- Entre 2015 e 2016 temos **TensorFlow** e **PyTorch** que trazem *deep learning* de modo acessível para a comunidade;



## Mas o Python não está sozinho ...

| Ecossistema | Curva | <i>Libs</i> | Comunidade | Performance |
|-------------|-------|-------------|------------|-------------|
| Python      | ↓     | ↑           | ↑          | ↓           |
| Julia       | ↓     | ↑           | ↓          | ●           |
| Mojo        | ↓     | ↑           | ↓          | ↑           |
| R           | ●     | ●           | ●          | ●           |
| Rust        | ↑     | ↓           | ↓          | ↑           |

## Ou seja ...

A combinação de um ecossistema de *libs* grande e abrangente, com uma comunidade numerosa, pujante e uma barreira de entrada na linguagem baixa (sintaxe simples, linguagem interpretada, etc.) **fazem do python o principal ecossistema para desenvolvimento de IA atualmente.**

Mesmo considerando suas fragilidades, como baixa de segurança de *runtime*, baixo desempenho e não ser nativa da *Web*.

# Conclusão

**Para entrar no mundo de IA eu preciso  
saber Python.**

**Ponto final!**

# Obrigado!

# E o ecossistema JS/TS?

## Dá para fazer *machine learning* com JavaScript?

Não apenas dá como temos um ecossistema cada vez mais completo e abrangente no que tange às tarefas para analisar dados, treinar modelos e entregar *data apps*.

## Machine Learning/ Deep Learning

- **TensorFlow.js**: dispensa apresentações. Portado diretamente do Python, pode inclusive converter modelos compilados em Python para rodar em JS. É acelerado por WebGPU e pode rodar no *browser*;
- **WebDNN**: biblioteca para executar redes neurais pré-compiladas diretamente no *browser*, com performance próxima a do servidor. Isso mesmo: exporte sua DNN do Keras ou qualquer outro *framework* e execute diretamente em um *React*, por exemplo;

## Gráficos e Plotagem

- **Plotly JS**: talvez uma das *lib* de visualização de dados mais populares no mundo Python tem seu *porting* para JavaScript;
- **D3.js**: a *lib de facto* quando se pensa em visualização de dados no mundo JS. É a base para diversas outras (inclusive o Plotly) e oferece diversidade e customização quase infinitas;



## Matemática e Estatística

- **Math.js**: biblioteca de matemática avançada com um *parser* de expressões bastante flexível. É praticamente um **Wolfram Alpha** dentro do seu código;
- **Numjs**: é o Numpy em JS. Faça de tudo um pouco com **NdArrays** ;

## Processamento de Linguagem Natural

- **Natural**: biblioteca com diversos utilitários para se trabalhar com linguagem natural, como tokenização, TF-IDF, N-gramas, distâncias entre *strings* e análise de sentimento;
- **Compromise**: biblioteca incrível para manipulação e tratamento de texto, com uma interface intuitiva e *blaaazing-faaast*;

## Dataframes e Data Wrangling

- **Danfo.js**: imagine que Pandas, Matplotlib e TensorFlow tem um filho: esse é o Danfo. É uma *lib* de alta performance para manipulação de dataframes e tensores e plotagem básica de dados;
- **Arquero**: *lib* bastante interessante para manipulação de dataframes, com uma sintaxe intuitiva e diversos utilitários que aumentam bastante a produtividade;
- **Pandas.js**: bom, o nome já diz tudo!

## Diferenciais do ecossistema JS/TS

O JS/TS traz features bastante interessantes para o jogo quando se pensa em desenvolvimento de aplicações de inteligência ou intensiva em dados:

- **Desempenho:** com WebGPU e WebAssembly, temos acesso a performance quase nativa;
- **Abrangência:** js é a linguagem mais popular do mundo. Em geral, é mais fácil ensinar um *framework* para uma pessoa que ensinar uma linguagem do zero;
- **Portabilidade:** Roda literalmente em qualquer lugar!

## Uma stack possível

- Deno com Jupyter Notebooks para análise de dados, data wrangling, aquisição de dados e treinamento do modelo.
- Deno com Fresh para criação de aplicações *web* intensivas em dados rápidas e fáceis de manter.
- Deno Deploy para publicar a aplicação na Edge para maximizar a experiência do usuário.

## WTF is Deno?



Deno rainha! Node nadinha!

## Desafios

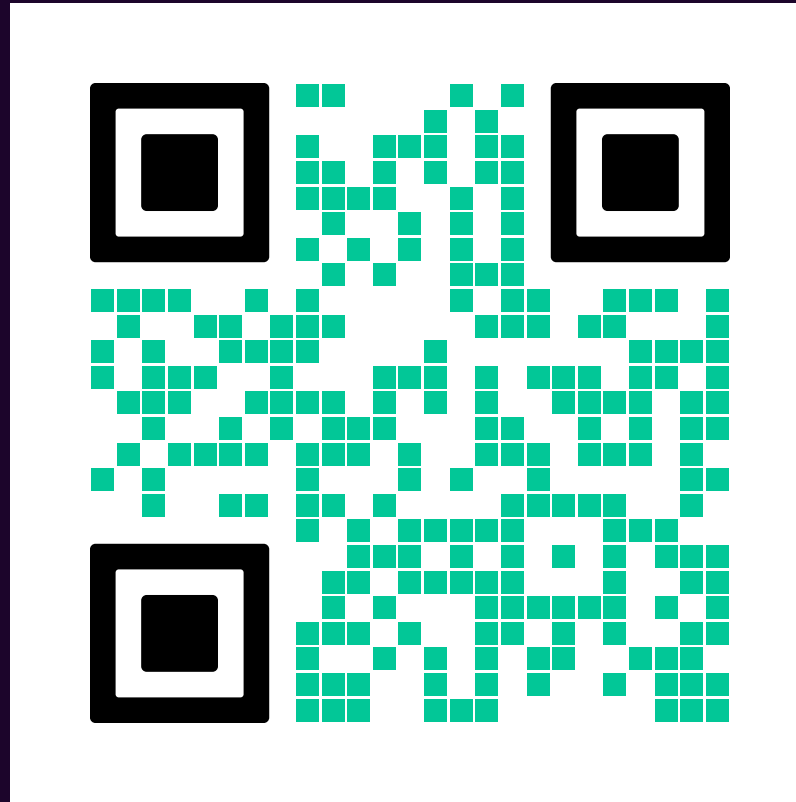
- Ecossistema insipiente;
- Comunidade pequena;
- Imagem não ser associada com essa categoria de problema;

## Vamos para a prática?

- *Data wrangling* com Jupyter, Deno e Danfo.js
- Machine Learning com TensorFlow.js



# Link do Repo



# Obrigado

- [linkedin.com/in/laion-azeredo](https://www.linkedin.com/in/laion-azeredo)
- <https://github.com/laionazeredo>