

# Ferramentas Científicas em Ruby

Amadeus Folego

<i>email</i>	amadeusfolego[at]gmail[dot]com
<i>github</i>	@badosu
<i>twitter</i>	@badosu_

- matplotlib

# Caso: Python/SciPy

- matplotlib
- numpy

# Caso: Python/SciPy

- matplotlib
- numpy
- ipython -pylab

- Prover estruturas numéricas

# NArray

- Prover estruturas numéricas
- <https://github.com/masa16/narray> (v0.6)

# NArray

- Prover estruturas numéricas
- <https://github.com/masa16/narray> (v0.6)
- Manipula valores em estruturas numéricas nativas do C

- Prover estruturas numéricas
- <https://github.com/masa16/narray> (v0.6)
- Manipula valores em estruturas numéricas nativas do C
- Dados: int (8,16,32), float (32,64) e complexo (64,128)



- Prover estruturas numéricas
- <https://github.com/masa16/narray> (v0.6)
- Manipula valores em estruturas numéricas nativas do C
- Dados: int (8,16,32), float (32,64) e complexo (64,128)
- Entre 28-50 vezes mais rápido

# NArray

- Prover estruturas numéricas
- <https://github.com/masa16/narray> (v0.6)
- Manipula valores em estruturas numéricas nativas do C
- Dados: int (8,16,32), float (32,64) e complexo (64,128)
- Entre 28-50 vezes mais rápido
- Consome 4 vezes menos memória, em média, do que Array

- Prover estruturas numéricas
- <https://github.com/masa16/narray> (v0.6)
- Manipula valores em estruturas numéricas nativas do C
- Dados: int (8,16,32), float (32,64) e complexo (64,128)
- Entre 28-50 vezes mais rápido
- Consome 4 vezes menos memória, em média, do que Array
- Pouco documentado, faltam features básicas

- Prover estruturas numéricas
- <https://github.com/masa16/narray> (v0.6)
- Manipula valores em estruturas numéricas nativas do C
- Dados: int (8,16,32), float (32,64) e complexo (64,128)
- Entre 28-50 vezes mais rápido
- Consome 4 vezes menos memória, em média, do que Array
- Pouco documentado, faltam features básicas
- Implementado em extensões em C, não compatível com JRuby

# NArray vs Array

```
# teste de performance  
n = 10**8
```

# NArray vs Array

```
# teste de performance  
n = 10**8  
  
# Array  
a = (1..n).to\_a  
b = (1..n).to\_a  
(1..n).map{|i| a[i]*b[i]}
```

# NArray vs Array

```
# teste de performance
```

```
n = 10**8
```

```
# Array
```

```
a = (1..n).to\_a
```

```
b = (1..n).to\_a
```

```
(1..n).map{|i| a[i]*b[i]}
```

```
# NArray
```

```
NArray.int(n).indgen * NArray.int(n).indgen
```

# NArray vs Array

```
# teste de performance
```

```
n = 10**8
```

```
# Array
```

```
a = (1..n).to\_a
```

```
b = (1..n).to\_a
```

```
(1..n).map{|i| a[i]*b[i]}
```

```
# NArray
```

```
NArray.int(n).indgen * NArray.int(n).indgen
```

- O segundo exemplo performou em 2.3s



# NArray vs Array

```
# teste de performance
```

```
n = 10**8
```

```
# Array
```

```
a = (1..n).to\_a
```

```
b = (1..n).to\_a
```

```
(1..n).map{|i| a[i]*b[i]}
```

```
# NArray
```

```
NArray.int(n).indgen * NArray.int(n).indgen
```

- O segundo exemplo performou em 2.3s
- O primeiro começou a travar a máquina após 4m30s

- Ambição de prover capacidade científica similar ao SciPy

- Ambição de prover capacidade científica similar ao SciPy
- Deve utilizar projetos como NArray e NMatrix em cima do ATLAS

- Ambição de prover capacidade científica similar ao SciPy
- Deve utilizar projetos como NArray e NMatrix em cima do ATLAS
- Última atualização há 6 meses

# Ferramentas Científicas em Ruby

Amadeus Folego

<i>email</i>	amadeusfolego[at]gmail[dot]com
<i>github</i>	@badosu
<i>twitter</i>	@badosu_