

# Ferramentas Científicas em Ruby

Amadeus Folego

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

# Porquê?

- Visualizar e extrair informações associadas a um grande conjunto de dados numéricos

# Porquê?

- Visualizar e extrair informações associadas a um grande conjunto de dados numéricos
- Manipulação de imagens, efetuar transformadas

# Porquê?

- Visualizar e extrair informações associadas a um grande conjunto de dados numéricos
- Manipulação de imagens, efetuar transformadas
- Simular modelos, otimizar parâmetros, minimizar funções

# Porquê?

- Visualizar e extrair informações associadas a um grande conjunto de dados numéricos
- Manipulação de imagens, efetuar transformadas
- Simular modelos, otimizar parâmetros, minimizar funções
- Inferir estatísticas

# Características Desejadas

- Manipulação de arrays numéricos de alta performance

# Características Desejadas

- Manipulação de arrays numéricos de alta performance
- REPL (Read Print Eval Loop) prático

# Características Desejadas

- Manipulação de arrays numéricos de alta performance
- REPL (Read Print Eval Loop) prático
- Ferramentas de visualização (plotting)



- Ferramenta de visualização: matplotlib

# Caso: Python/SciPy

- Ferramenta de visualização: matplotlib
- Manipulação numérica: numpy

# Caso: Python/SciPy

- Ferramenta de visualização: matplotlib
- Manipulação numérica: numpy
- REPL: 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

- Ambição de prover capacidade científica similar ao SciPy
- Deve utilizar projetos como NArray e NMatrix em cima do ATLAS
- Última atualização há 6 meses
- Projeto muito ambicioso, mas com bons frutos:

- Ambição de prover capacidade científica similar ao SciPy
- Deve utilizar projetos como NArray e NMatrix em cima do ATLAS
- Última atualização há 6 meses
- Projeto muito ambicioso, mas com bons frutos:
  - Visualization: rubyvis

- Ambição de prover capacidade científica similar ao SciPy
- Deve utilizar projetos como NArray e NMatrix em cima do ATLAS
- Última atualização há 6 meses
- Projeto muito ambicioso, mas com bons frutos:
  - Visualization: rubyvis
  - Statistics: statsample

- Ambição de prover capacidade científica similar ao SciPy
- Deve utilizar projetos como NArray e NMatrix em cima do ATLAS
- Última atualização há 6 meses
- Projeto muito ambicioso, mas com bons frutos:
  - Visualization: rubyvis
  - Statistics: statsample
  - Numeric: minimization, integration, nmatrix

- Ambição de prover capacidade científica similar ao SciPy
- Deve utilizar projetos como NArray e NMatrix em cima do ATLAS
- Última atualização há 6 meses
- Projeto muito ambicioso, mas com bons frutos:
  - Visualization: rubyvis
  - Statistics: statsample
  - Numeric: minimization, integration, nmatrix
  - Probability: distribution



- Operações em conjuntos ordenados

- Operações em conjuntos ordenados
- Único requisito é incluir Comparable, e compatibilidade numérica para alguns métodos

- Operações em conjuntos ordenados
- Único requisito é incluir Comparable, e compatibilidade numérica para alguns métodos

- Operações em conjuntos ordenados
- Único requisito é incluir Comparable, e compatibilidade numérica para alguns métodos

```
Interval[1,5] | Interval[7,9] == Interval[[1,5],[7,9]]
```

- Operações em conjuntos ordenados
- Único requisito é incluir Comparable, e compatibilidade numérica para alguns métodos

```
Interval[1,5] | Interval[7,9] == Interval[[1,5],[7,9]]
```

```
Interval['a','f'] ^ Interval['c','g'] == Interval['c','f']
```

# Ferramentas Científicas em Ruby

Amadeus Folego

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