INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Programação de Computadores

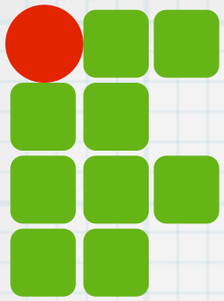
Exercícios de avaliação

O que veremos hoje?

* Como testar seu programa
  * Criação de arquivos de teste
  * Redirecionamento de entrada/saída

Teste

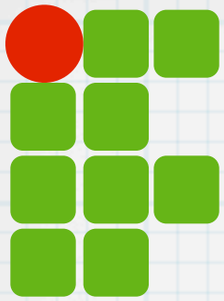
* Ações que permitem verificar corretude do programa
  * Importante sistematizar
  * Tarefa repetitiva: pode ser automatizada
* Testes automáticos
  * Facilita
  * Acelera
  * Usa mesmo conjunto de testes
    * Teste de não regressão
    * Conjunto pode aumentar com novas funcionalidades



Como fazer?

* Método mais simples
  * Criar arquivos com os dados de entrada
  * Executar os testes e gerar arquivos com os dados de saída
  * Verificar se a saída está de acordo com o que esperamos

Como fazer isso?



Redirecionamento

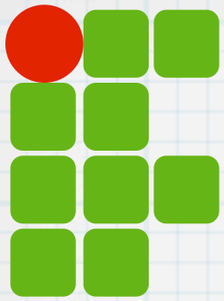
* Quando executamos um programa
  * O comando `gets`
    * Espera dados do teclado
  * Os comandos `puts` e `print`
    * Mostram os dados na tela
* Redirecionamento
  * Entrada: Os dados devem ser lidos de um arquivo ao invés do teclado
  * Saída: Os dados devem ser "mostrados" em um arquivo ao invés da tela

Redirecionamento

* Saída
  * Deve ser feito na execução do programa
    * `ruby programa.rb > saida.txt`
    * A execução desse programa cria o arquivo saida.txt contendo os caracteres "impressos" pelo programa (`puts` e `print`)

oi.rb

```
puts "Meu primeiro programa!"
puts "Segunda linha do meu primeiro programa!"
```



Redirecionamento

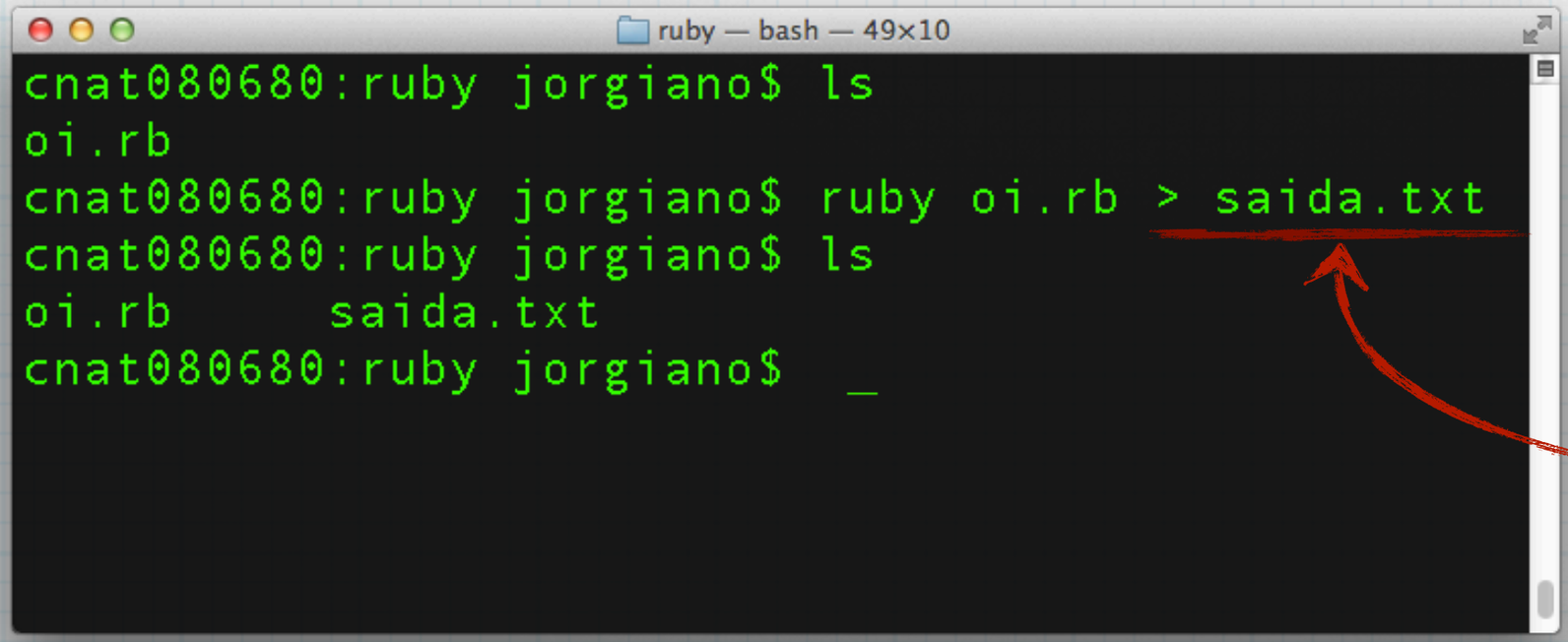
oi.rb

```
puts "Meu primeiro programa!"
puts "Segunda linha do meu primeiro programa!"
```

Redirecionamento

oi.rb

```
puts "Meu primeiro programa!"
puts "Segunda linha do meu primeiro programa!"
```



O programa vai "escrever" os dados no arquivo saida.txt

saida.txt

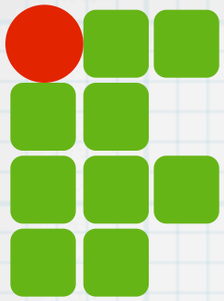
```
Meu primeiro programa!
Segunda linha do meu primeiro programa!
```

Redirecionamento

- Entrada
  - Deve ser feito na execução do programa
    - `ruby programa.rb < entrada.txt`
    - A execução desse programa precisa do arquivo entrada.txt contendo os caracteres que servem de entrada para o programa (`gets`)

media.rb

```ruby
nota1=gets.to_f
nota2=gets.to_f
media=(nota1*2+nota2*3)/5
puts "%.2f" % media
```



Redirecionamento

media.rb

```
nota1=gets.to_f
nota2=gets.to_f
media=(nota1*2+nota2*3)/5
puts "%.2f" % media
```

Redirecionamento

media.rb

```
nota1=gets.to_f
nota2=gets.to_f
media=(nota1*2+nota2*3)/5
puts "%.2f" % media
```

entrada.txt

```
8.8
9.1
```

```
ruby — bash — 55×7
cnat080680:ruby jorgiano$ ls
entrada.txt media.rb
cnat080680:ruby jorgiano$ ruby media.rb < entrada.txt
8.98
cnat080680:ruby jorgiano$ _
```

O programa vai "ler" os dados do arquivo entrada.txt

Redirecionamento

* Entrada e saída

media.rb

```
nota1=gets.to_f
nota2=gets.to_f
media=(nota1*2+nota2*3)/5
puts "%.2f" % media
```

entrada.txt

```
8.8
9.1
```

```
ruby — bash — 66×5
cnat080680:ruby jorgiano$ ls
entrada.txt media.rb
cnat080680:ruby jorgiano$ ruby media.rb < entrada.txt > saida.txt
cnat080680:ruby jorgiano$ ls
entrada.txt media.rb    saida.txt
```

O programa executa lendo os dados de `entrada.txt` e cria o arquivo `saida.txt` com os dados de saída

saida.txt

```
8.98
```

Para testar

* Crie o arquivo com os dados de entrada
* Desenvolva o programa
* Execute e gere o arquivo de saída
* Verifique se conteúdo do arquivo de saída é o esperado

Dúvidas?