

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

Programação de Computadores

Exercícios de avaliação

Copyright © 2012 IFRN



O que veremos hoje?

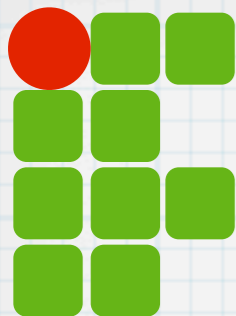
- * Sistema de envio de exercícios
 - * Página WEB
 - * O que informar
- * Como testar seu programa
 - * Criação de arquivos de teste
 - * Redirecionamento de entrada/saída





Introdução

- * Listas de exercícios
 - * Pontos para avaliação
 - * Atividades extra-classe
 - * Devem ser enviadas através da WEB



Lista 00

IFRN/TADS
Programação de computadores
Professor: Jorgiano

***Obrigatório**

Matricula *

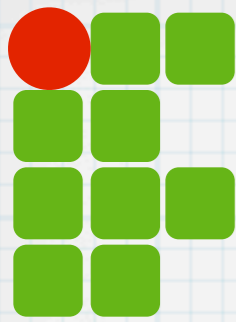
Numero *

Resposta *

Enviar

Tecnologia do [Google Docs](#)

[Denunciar abuso](#) - [Termos de Utilização](#) - [Termos adicionais](#)



Informações
da lista

Matrícula
(sem pontos,
apenas
números)

Número da
questão
(APENAS o
número)

Programa
escrito em
ruby

Lista 00

IFRN/TADS
Programação de computadores
Professor: Jorgiano

*Obrigatório

Matricula *

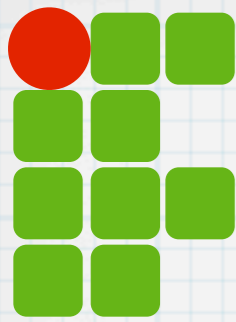
Numero *

Resposta *

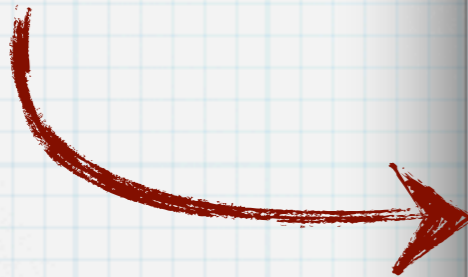
Enviar

Tecnologia do [Google Docs](#)

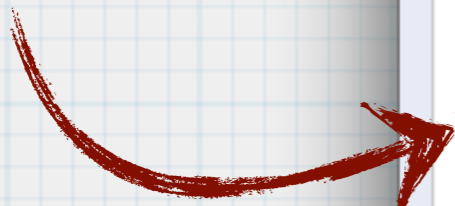
[Denunciar abuso](#) - [Termos de Utilização](#) - [Termos adicionais](#)



Verifique se
está correto



Clica em
enviar



Lista 00

IFRN/TADS
Programação de computadores
Professor: Jorgiano

***Obrigatório**

Matricula*
20121014050000

Numero*
004

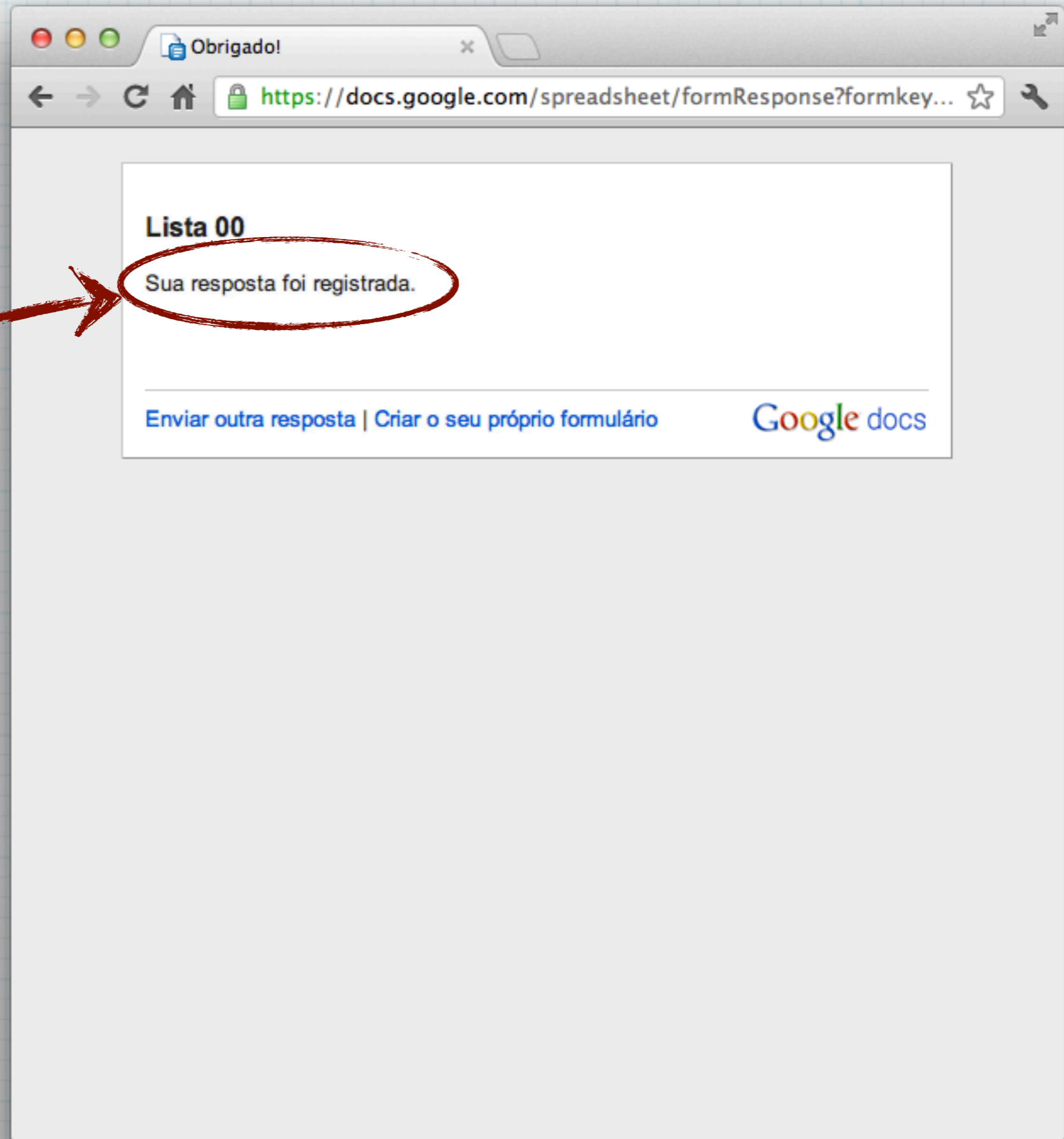
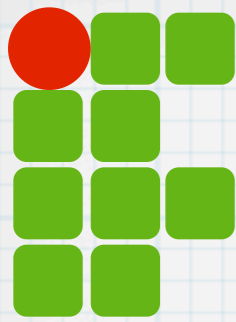
Resposta *

```
n1 = gets.to_f  
n2 = gets.to_f  
media = (n1+n2)/2  
  
puts "%.2f" % media
```

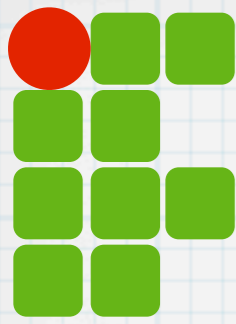
Enviar

Tecnologia do [Google Docs](#)

[Denunciar abuso](#) - [Termos de Utilização](#) - [Termos adicionais](#)



**Verifique se a
submissão foi
registrada**



Importante

- * **Pode enviar o mesmo exercício várias vezes**
 - * Apenas o último a ser enviado é considerado
- * **Após a data limite não é possível enviar mais respostas**
- * **As listas são INDIVIDUAIS!!!**





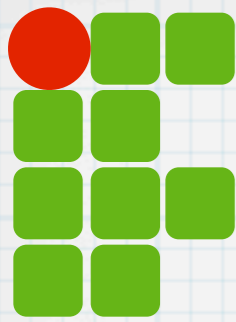
Teste

* Antes de enviar a resposta:

- * Teste o programa (várias vezes, com vários valores de entrada)

* Automatizar os testes

- * Facilita
- * Acelera
- * Usa mesmo conjunto de testes



Como fazer?

* Modelo mais simples

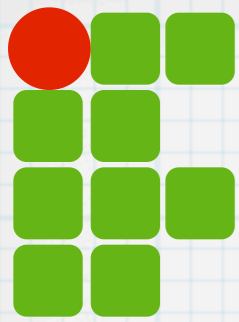
- * Criar arquivos com os dados de entrada
- * Executar os testes e gerar arquivos com os dados de saída
- * Verificar se a saída está de acordo com o que esperamos

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!"
```

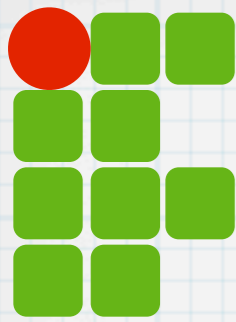
```
puts "Segunda linha do meu primeiro programa!"
```




Redirecionamento

oi.rb

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



Redirecionamento

oi.rb

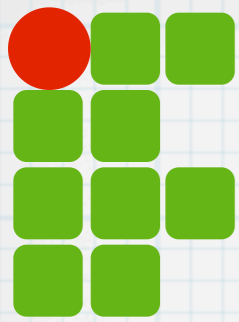
```
puts "Meu primeiro programa!"  
puts "Segunda linha do meu primeiro programa!"
```

```
ruby — bash — 49x10  
cnat080680:ruby jorgiano$ ls  
oi.rb  
cnat080680:ruby jorgiano$ ruby oi.rb > saida.txt  
cnat080680:ruby jorgiano$ ls  
oi.rb      saida.txt  
cnat080680:ruby jorgiano$ _
```

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`

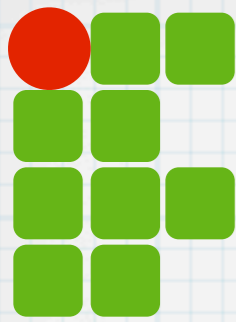
```
nota1=gets.to_f
nota2=gets.to_f
media=(nota1*2+nota2*3)/5
puts "%.2f" % media
```




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
```

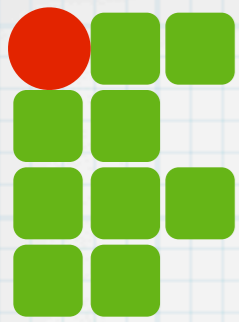
entrada.txt

```
8.8  
9.1
```

```
ruby -- bash -- 55x7  
cnat080680:ruby jorgiano$ ls  
entrada.txt media.rb  
cnat080680:ruby jorgiano$ ruby media.rb < entrada.txt  
8.98  
cnat080680:ruby jorgiano$ _
```

A red arrow points from the text below to the redirection operator in the terminal command.

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 -- 66x5  
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

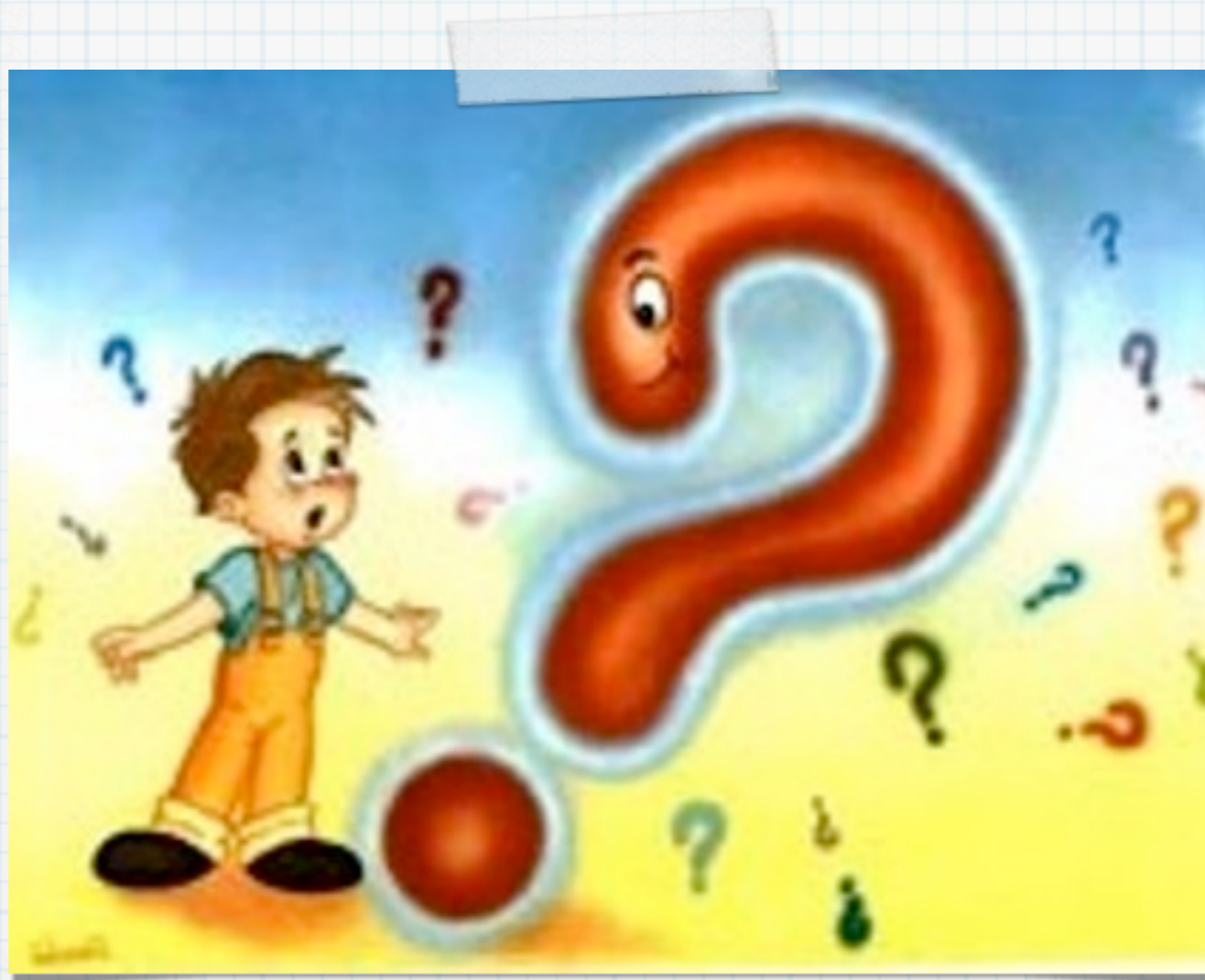
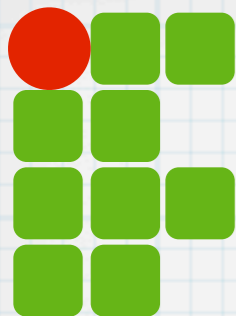
saida.txt

```
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?