

# Laboratorio de Computación IV

## Clase 8

*Andrés Fortier*

# ¿Consultas?

---

- HTTP.
- HTML.



# Comando del día: cat (y redirect)

- Los programas manejan tres streams: STDIN, STDOUT, STDERR.
- Hasta ahora vimos
  - STDIN: Teclado.
  - STDOUT y STDERR: Consola.
- Linux provee redireccionar utilizando los operadores “<” y “>”

# Comando del día: cat (y redirect)

```
$ ls -la
```

```
$ ls -la > salida.txt
```

- Veán el contenido de `salida.txt`.

```
top -n 1 | grep Mem > salida.txt
```

- Veán nuevamente el contenido de `salida.txt`.

# Comando del día: cat

- cat == ver/copiar/concatenar archivos.
- Ver el contenido de un archivo

```
$ cat salida.txt
KiB Mem:    8166488 total,  3860712 used,  4305776 free...
KiB Swap: 13056636 total,      0 used, 13056636 free...
```

- Copiar el contenido de un archivo a otro

```
$ cat salida.txt > copia_de_salida.txt
$ cat copia_de_salida.txt
KiB Mem:    8166488 total,  3860712 used,  4305776 free...
KiB Swap: 13056636 total,      0 used, 13056636 free...
```

# Comando del día: cat

- Concatenar archivos
  - Crear `texto.txt` con contenido “Hola”

```
$ cat salida.txt texto.txt
KiB Mem:  8166488 total, 3860712 used, 4305776 free...
KiB Swap: 13056636 total, 0 used, 13056636 free...
Hola
```

```
$ cat salida.txt texto.txt > combinado.txt
$ cat combinado.txt
```

# Comando del día: cat

- Crear archivos

```
$ cat > nuevo.txt  
Hola mundo
```

```
CTRL + D
```

```
$ cat nuevo.txt  
Hola mundo
```

- -n → enumerar líneas

```
$ cat -n salida.txt  
1  KiB Mem:    8166488 total,   3860712 used,   4305776...  
2  KiB Swap: 13056636 total,         0 used, 13056636...
```

# rspec

- Identidad vs igualdad

```
it "Compares identity vs equality" do
  i1 = "Hola"
  i2 = "Hola"
  expect(i1).to eq(i2) (Ok)
  expect(i1).to be(i2) (Fail)
end
```

```
it "Compares identity vs equality" do
  i1 = "Hola"
  i2 = "Hola"
  expect(i1).to eq(i2)
  expect(i1).not_to be(i2)
  expect(i1).to be(i1)
end
```



# rspec

- Built-in matchers

```
it "Shows some expectation examples" do
  expect(5).to be > 4
  expect(5).to be >= 5
  expect(5).to be <= 7
  expect(5).to be < 6
  expect(5).to be_between(1, 5).inclusive
  expect(5).to be_between(1, 6).exclusive
  expect(4.5).to be_within(0.2).of(4.6)

  expect("Hola").to start_with "Ho"
  expect("Hola").to end_with "la"

  expect(1 == 1).to be true
  expect(1 == 2).to be false

  expect(@variable).to be_nil
end
```

# rspec

- Excepciones

```
it "Raises an error" do
  expect {
    raise(ArgumentError, "Mensaje!")
  }.to raise_error

  expect {
    raise(ArgumentError, "Mensaje!")
  }.to raise_error(ArgumentError)

  expect {
    raise(ArgumentError, "Mensaje!")
  }.to raise_error(ArgumentError, "Mensaje!")
end
```

# rspec

- Inclusión en colecciones

```
it "Tests collections" do
  expect([1, 2, 3]).to      include(1)
  expect([1, 2, 3]).to      include(1, 2)
  expect("this string").to  include("is str")
  expect([1, 2, 3]).to      match_array([3, 2, 1])
end
```

# rspec

- Specs pendientes

```
it "Encrypts password using bcrypt with 10 iterations"
```

```
.....*
```

```
Pending: (Failures listed here are expected and do not  
affect your suite's status)
```

```
  1) Some tests Encrypts password using bcrypt with 10  
iterations
```

```
    # Not yet implemented
```

```
    # ./spec/test_spec.rb:54
```

```
Finished in 0.00675 seconds (files took 0.11865 seconds  
to load)
```

```
7 examples, 0 failures, 1 pending
```

# rspec

- Setup / teardown

```
before :all do
  puts "Al comienzo de la suite de test"
end

before :each do
  puts "Antes de cada test"
end

after :each do
  puts "Al finalizar cada test"
end

after :all do
  puts "Al final de la suite de test"
end
```

# rspec

- Setup / teardown

**Al comienzo de la suite de test**

Antes de cada test

Al finalizar cada test

.

Antes de cada test

Al finalizar cada test

.

Antes de cada test

Al finalizar cada test

.

Antes de cada test

Al finalizar cada test

.

**Al final de la suite de test**

# rspec

- let

```
it "Has a value 0 after creation" do
  expect(Contador.new.valor).to be(0)
end

it "Increments" do
  expect(Contador.new.siguiente).to be(1)
end
```

# rspec

- let

```
before :each do
  @contador = Contador.new
end

it "Has a value 0 after creation" do
  expect(@contador.valor).to be(0)
end

it "Increments" do
  expect(@contador.siguiente).to be(1)
end
```



# rspec

- let

```
let (:contador) {Contador.new}

it "Has a value 0 after creation" do
  expect(contador.valor).to be(0)
end

it "Increments" do
  expect(contador.siguiente).to be(1)
end
```

# rspec

---

- Let
  - Se evalúa una vez por cada test.
  - Sólo si se usa (*lazy*).
  - Se cachea una vez evaluado.