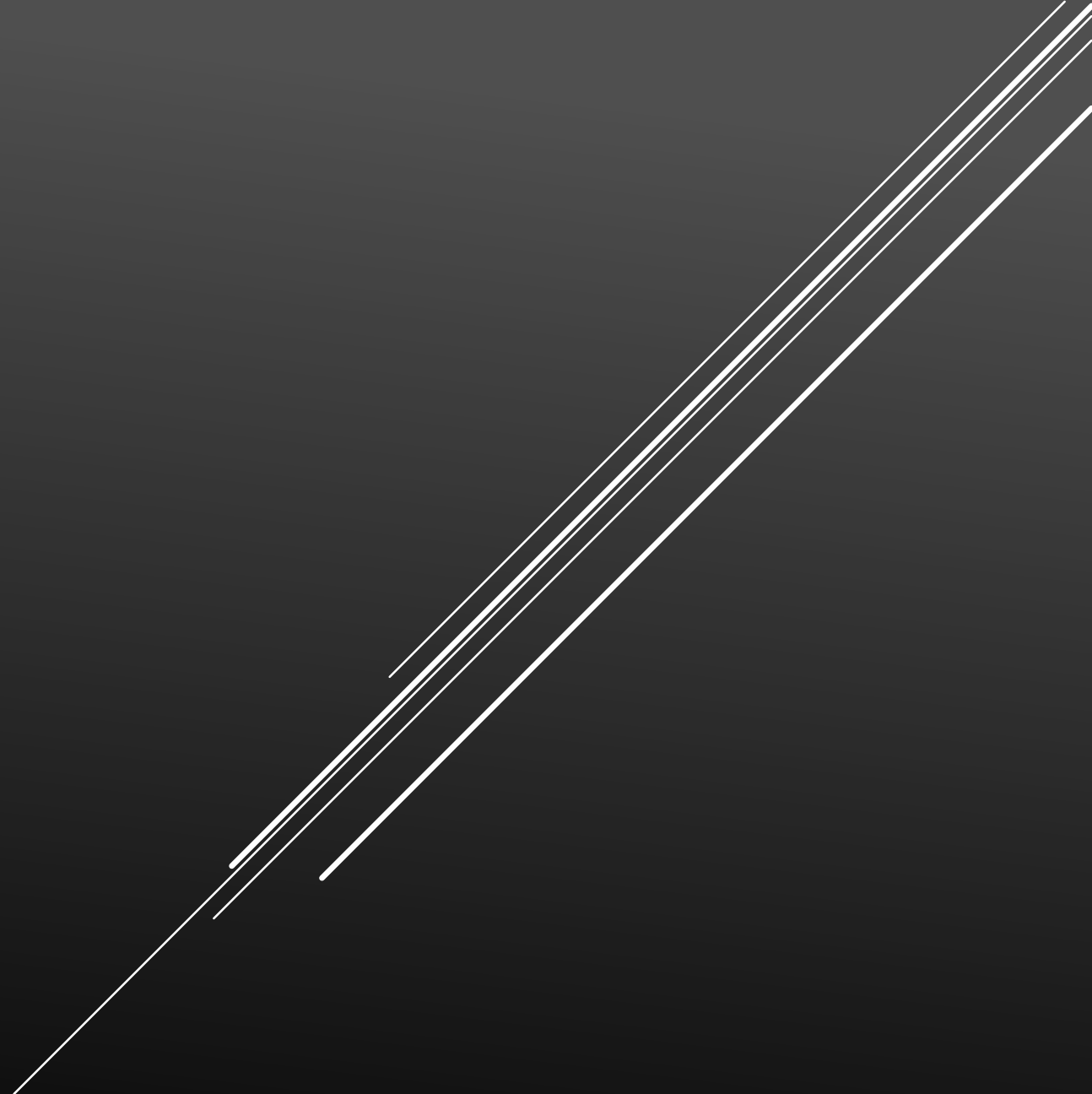


# RUBY ON RAILS

Élisson Michael F. M. Araújo



# TABELA DE TWEETS

## 3 COLUNAS

## 4 LINHAS


tweets		
	A	B
1	Where can I get a good bite to eat?	Ash
2	My left arm is missing, but I don't care.	Bob
3	I just ate some delicious brains.	Jim
4	OMG, my fingers turned green. #FML	Ash

# NOME DAS COLUNAS

id

status

zombie



	A	B
1	Where can I get a good bite to eat?	Ash
2	My left arm is missing, but I don't care.	Bob
3	I just ate some delicious brains.	Jim
4	OMG, my fingers turned green. #FML	Ash


# DESAFIO 1:

## OBTER O TWEET COM ID = 3

id

status

zombie



	A	B
1	Where can I get a good bite to eat?	Ash
2	My left arm is missing, but I don't care.	Bob
3	I just ate some delicious brains.	Jim
4	OMG, my fingers turned green. #FML	Ash

# DESAFIO 1:

## OBTER UM *HASH* DO TWEET COM ID = 3

### RESULTADO QUE QUEREMOS:

```
hash = { :status => "I just ate some delicious brains",  
         :zombie => "Jim" }
```

id		status	zombie
		A	B
1		Where can I get a good bite to eat?	Ash
2		My left arm is missing, but I don't care.	Bob
3		I just ate some delicious brains.	Jim
4		OMG, my fingers turned green. #FML	Ash

# HASH – UMA SÉRIE DE PARES DE CHAVES E VALORES

```
hash = { :status => "I just ate some delicious brains",  
         :zombie => "Jim" }
```

# HASH – UMA SÉRIE DE PARES DE CHAVES E VALORES

```
hash = { :status => "I just ate some delicious brains",  
         :zombie => "Jim" }
```

```
puts hash[:status]
```

```
"I just ate some delicious brains"
```

Several white lines of varying lengths and angles are drawn in the bottom right corner of the slide, creating a decorative graphic element.

# HASH – UMA SÉRIE DE PARES DE CHAVES E VALORES

```
hash = { :status => "I just ate some delicious brains",  
         :zombie => "Jim" }
```

```
puts hash[:status]
```

"I just ate some delicious brains"

```
puts hash[:zombie]
```

"Jim"

Several white lines of varying lengths and angles are drawn in the bottom right corner of the slide, creating a decorative graphic element.



# HASH – UMA SÉRIE DE PARES DE CHAVES E VALORES

```
hash = { :status => "I just ate some delicious brains",  
         :zombie => "Jim" }
```

```
puts hash[:status]
```

"I just ate some delicious brains"

```
puts hash[:zombie]
```

"Jim"

```
puts hash[:zombie] + " said " + hash[:status]
```

"Jim said I just ate some delicious brains"

# tweets

id	status	zombie
1	Where can I get a good bite to eat?	Ash
2	My left arm is missing, but I don't care.	Bob
3	I just ate some delicious brains.	Jim
4	OMG, my fingers turned green. #FML	Ash

DESAFIO 1:  
OBTENHA UM *HASH* DO TWEET COM ID = 3  
RESPOSTA:

```
t = Tweet.find(3)
```

# tweets

id	status	zombie
1	Where can I get a good bite to eat?	Ash
2	My left arm is missing, but I don't care.	Bob
3	I just ate some delicious brains.	Jim
4	OMG, my fingers turned green. #FML	Ash

```
t = Tweet.find(3)
```

```
puts t[:id]      3
```

```
puts t[:status]  "I just ate some delicious brains."
```

```
puts t[:zombie]  "Jim"
```

```
t = Tweet.find(3)
```

```
puts t[:id]
```

```
puts t[:status]
```

```
puts t[:zombie]
```

Mesmo  
Resultado

```
puts t.id
```

```
puts t.status
```

```
puts t.zombie
```

```
t = Tweet.find(3)
```

*Pluralize*

**tweets**

id	status	zombie
1	Where can I get a good bite to eat?	Ash
2	My left arm is missing, but I don't care.	Bob
3	I just ate some delicious brains.	Jim
4	OMG, my fingers turned green. #FML	Ash

**Create**

**Read**

**Update**

**Delete**

**Create**

```
t = Tweet.new  
t.status = "I <3 brains."  
t.save
```

**Read**

**Update**

**Delete**

## Create

```
t = Tweet.new  
t.status = "I <3 brains."  
t.save
```

## Read

```
Tweet.find(3)
```

## Update

## Delete



## Create

```
t = Tweet.new  
t.status = "I <3 brains."  
t.save
```

## Read

```
Tweet.find(3)
```

## Update

```
t = Tweet.find(3)  
t.zombie = "EyeballChomper"  
t.save
```

## Delete

## Create

```
t = Tweet.new  
t.status = "I <3 brains."  
t.save
```

## Read

```
Tweet.find(3)
```

## Update

```
t = Tweet.find(3)  
t.zombie = "EyeballChomper"  
t.save
```

## Delete

```
t = Tweet.find(3)  
t.destroy
```

## Create

```
t = Tweet.new  
t.status = "I <3 brains."  
t.save
```

## Read

```
Tweet.find(3)
```

## Update

```
t = Tweet.find(3)  
t.zombie = "EyeballChomper"  
t.save
```

## Delete

# SINTAXE DO *CREATE* (CRIAR)

```
t = Tweet.new  
t.status = "I <3 brains."  
t.zombie = "Jim"  
t.save
```

Observe que não  
precisamos atribuir  
um id.

# SINTAXE DO *CREATE* (CRIAR)

```
t = Tweet.new  
t.status = "I <3 brains."  
t.zombie = "Jim"  
t.save
```

Observe que não  
precisamos atribuir  
um id.

## SINTAXE ALTERNATIVA

```
t = Tweet.new(:status => "I <3 brains", :zombie => "Jim")  
t.save
```

# SINTAXE DO CREATE (CRIAR)

```
t = Tweet.new  
t.status = "I <3 brains."  
t.zombie = "Jim"  
t.save
```

Observe que não precisamos atribuir um id.

## SINTAXES ALTERNATIVAS

```
t = Tweet.new(:status => "I <3 brains", :zombie => "Jim")  
t.save
```

```
Tweet.create(:status => "I <3 brains", :zombie => "Jim")
```

# SINTAXE DO *READ*(LER)

`Tweet.find(2)`

Retorna um único item

`Tweet.find(3, 4, 5)`

Retorna um vetor (*array*)

`Tweet.first`

Retorna o primeiro item

`Tweet.last`

Retorna o último item

`Tweet.all`

Retorna todos os itens

`Tweet.count`

Retorna a contagem de itens



# SINTAXE DO *READ(LER)*

`Tweet.count`

Retorna a contagem de itens

`Tweet.order(:zombie)`

Todos ordenados por zumbi

`Tweet.limit(10)`

Apenas 10 itens

`Tweet.where(:zombie => "ash")`

Apenas tweets de Ash





# SINTAXE DO *READ(ER)*

`Tweet.count`

Retorna a contagem de itens

`Tweet.order(:zombie)`

Todos ordenados por zumbi

`Tweet.limit(10)`

Apenas 10 itens

`Tweet.where(:zombie => "ash")`

Apenas tweets de Ash

Podemos Combinar Todos

`Tweet.where(:zombie => "ash").order(:zombie).limit(10)`

*method chaining*

# SINTAXE DO *UPDATE* (ATUALIZAR)

```
t = Tweet.find(3)
t.zombie = "EyeballChomper"
t.save
```

# SINTAXE DO *UPDATE* (ATUALIZAR)

```
t = Tweet.find(3)
t.zombie = "EyeballChomper"
t.save
```

## SINTAXE ALTERNATIVA

```
t = Tweet.find(2)
t.attributes = {
  :status => "Can I munch your eyeballs?",
  :zombie => "EyeballChomper"
}

t.save
```

# SINTAXES ALTERNATIVAS DO UPDATE

```
t = Tweet.find(2)
t.attributes = {
  :status => "Can I munch your eyeballs?",
  :zombie => "EyeballChomper"
}

t.save
```

```
t = Tweet.find(2)
t.update_attributes(
  :status => "Can I munch your eyeballs?",
  :zombie => "EyeballChomper"
)
```

# SINTAXE DO *DELETE* (EXCLUIR)

```
t = Tweet.find(2)  
t.destroy
```

```
Tweet.find(2).destroy
```

```
Tweet.destroy_all
```