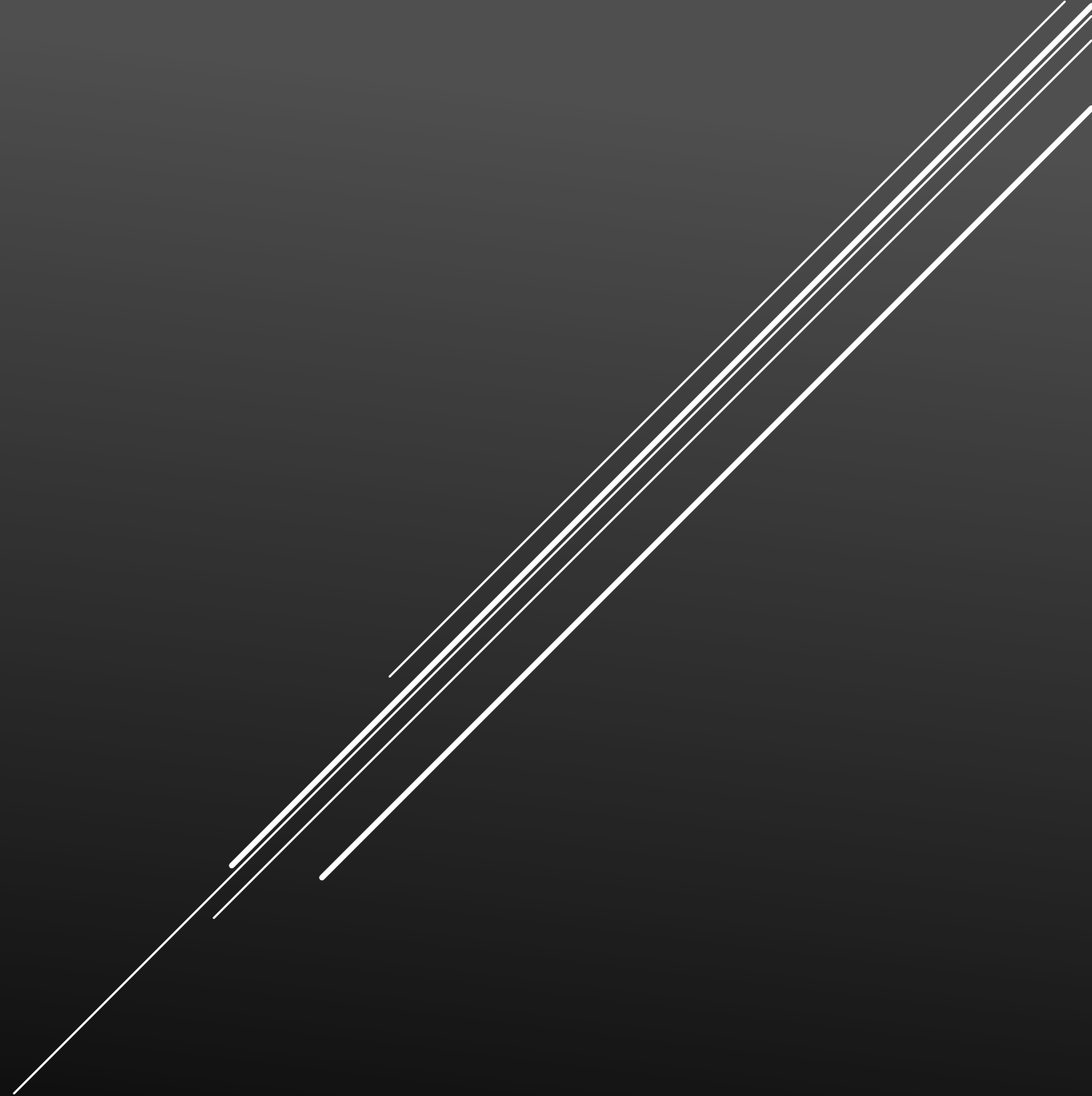


RUBY ON RAILS

Élisson Michael F. M. Araújo



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



Como o
Mapeamento
acontece?

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

NO SLIDE
ANTERIOR

MODELOS (MODELS)

Tweet

app/models/tweet.rb

```
class Tweet < ActiveRecord::Base  
  
end
```

MODELOS (MODELS)

Tweet

app/models/tweet.rb

```
class Tweet < ActiveRecord::Base  
  
end
```

Mapeia a Classe para a Tabela

Tweet

MODELOS (MODELS)

```
app/models/tweet.rb  
class Tweet < ActiveRecord::Base  
end
```

Mapeia a Classe para a Tabela

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

MODELOS (MODELS)

```
app/models/tweet.rb
```

```
class Tweet < ActiveRecord::Base  
  
end
```

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

classe

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

MODELOS (MODELS)

```
app/models/tweet.rb  
class Tweet < ActiveRecord::Base  
  
end
```

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

classe

find é um método
de classe

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

MODELOS (MODELS)

```
app/models/tweet.rb
```

```
class Tweet < ActiveRecord::Base  
  
end
```

objeto/instância

Tweet #3

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

classe

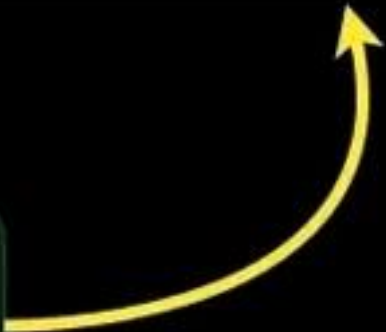
tweets

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

VALIDAÇÕES (VALIDATIONS)

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
5		

```
t = Tweet.new  
t.save
```



VALIDAÇÕES (VALIDATIONS)

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
5		

```
t = Tweet.new  
t.save
```

É possível criar um
tweet sem
conteúdo

VALIDAÇÕES (VALIDATIONS)

```
app/models/tweet.rb  
class Tweet < ActiveRecord::Base  
  
  validates_presence_of :status  
  
end
```

```
app/models/tweet.rb
```

```
class Tweet < ActiveRecord::Base  
  
  validates_presence_of :status  
  
end
```

```
> t = Tweet.new
```

```
=> #<Tweet id: nil, status: nil, zombie: nil>
```

```
app/models/tweet.rb
```

```
class Tweet < ActiveRecord::Base  
  
  validates_presence_of :status  
  
end
```

```
> t = Tweet.new
```

```
=> #<Tweet id: nil, status: nil, zombie: nil>
```

```
> t.save
```

```
=> false
```

```
app/models/tweet.rb
```

```
class Tweet < ActiveRecord::Base  
  
  validates_presence_of :status  
  
end
```

```
> t = Tweet.new
```

```
=> #<Tweet id: nil, status: nil, zombie: nil>
```

```
> t.save
```

```
=> false
```

```
> t.errors
```

```
=> {:status=>["can't be blank"]}
```



```
app/models/tweet.rb
```

```
class Tweet < ActiveRecord::Base  
  
  validates_presence_of :status  
  
end
```

```
> t = Tweet.new  
      => #<Tweet id: nil, status: nil, zombie: nil>  
  
> t.save  
      => false  
  
> t.errors  
      => {:status=>["can't be blank"]}  
  
> t.errors[:status]  
      => ["can't be blank"]
```


Validações (Validations)

`validates_presence_of :status`

`validates_numericality_of :fingers`

`validates_uniqueness_of :toothmarks`

`validates_confirmation_of :password`

`validates_acceptance_of :zombification`

`validates_length_of :password, :minimum => 3`

`validates_format_of :email, :with => /regex/i`

`validates_inclusion_of :age, :in => 21..99`

`validates_exclusion_of :age, :in => 0...21,
:message => "Sorry you must be over 21"`

`validates_presence_of :status`

Está Prenchido?

`validates_numericality_of :fingers`

É um Número?

`validates_uniqueness_of :toothmarks`

É Único?

`validates_confirmation_of :password`

Foi Confirmado?

`validates_acceptance_of :zombification`

Foi Aceito?

`validates_length_of :password, :minimum => 3`

Tamanho

`validates_format_of :email, :with => /regex/i`

Formato

`validates_inclusion_of :age, :in => 21..99`

Inclusão

`validates_exclusion_of :age, :in => 0...21,`

Exclusão

`:message => "Sorry you must be over 21"`

VALIDAÇÕES (VALIDATIONS)

```
app/models/tweet.rb
```

```
class Tweet < ActiveRecord::Base  
  
  validates_presence_of :status  
  
end
```

Atributo

Validação

```
validates :status, :presence => true
```

VALIDAÇÕES (VALIDATIONS)

Atributo

Validação

validates :status, :presence => true

validates :status, :length => { :minimum => 3 }

Atributo

Validação

Validações
Nova Sintaxe

```
validates :status, :presence => true  
validates :status, :length => { :minimum => 3 }
```

```
validates :status, :presence => true, :length => { :minimum => 3 }
```

VALIDAÇÕES NOVA SINTAXE

```
:presence => true
:uniqueness => true
:numericality => true
:length => { :minimum => 0, :maximum => 2000 }
:format => { :with => /.*/ }
:inclusion => { :in => [1,2,3] }
:exclusion => { :in => [1,2,3] }
:acceptance => true
:confirmation => true
```

RELACIONAMENTOS (RELATIONSHIPS)

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



Repetição



Repetição

RELACIONAMENTOS (*RELATIONSHIPS*)

tweets

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

zombies

id	name	graveyard
1	Ash	Glen Haven Memorial Cemetery
2	Bob	Chapel Hill Cemetery
3	Jim	My Father's Basement

RELACIONAMENTOS (RELATIONSHIPS)

tweets

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

Como Relacionar?

zombies

id	name	graveyard
1	Ash	Glen Haven Memorial Cemetery
2	Bob	Chapel Hill Cemetery
3	Jim	My Father's Basement



RELACIONAMENTOS (RELATIONSHIPS)

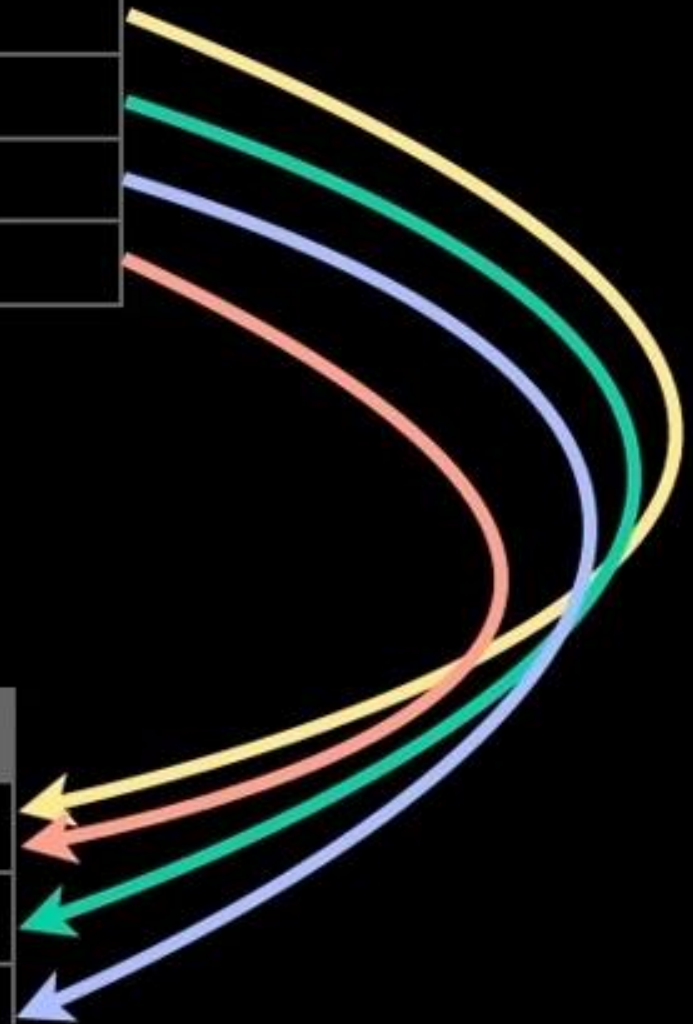
tweets

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

Atenção p/ Convenção

zombies

id	name	graveyard
1	Ash	Glen Haven Memorial Cemetery
2	Bob	Chapel Hill Cemetery
3	Jim	My Father's Basement



RELACIONAMENTOS (RELATIONSHIPS)

tweets

id	status	zombie_id
1	Where can I get a good bite to eat?	1
2	My left arm is missing, but I don't care.	2
3	I just ate some delicious brains.	3
4	OMG, my fingers turned green. #FML	1
5	Your eyelids taste like bacon.	2

zombies

id	name	graveyard
1	Ash	Glen Haven Memorial Cemetery
2	Bob	Chapel Hill Cemetery
3	Jim	My Father's Basement



RELACIONAMENTOS (*RELATIONSHIPS*)

Um tweet pertence
a um zumbi

Um zumbi tem
muitos tweets

app/models/tweet.rb

```
class Tweet < ActiveRecord::Base  
  belongs_to :zombie  
end
```

Singular



app/models/zombie.rb

```
class Zombie < ActiveRecord::Base  
  has_many :tweets  
end
```

Plural



Um tweet pertence
a um zumbi

```
> z = Zombie.find(2)  
=> #<Zombie id: 2, name: "Bob", graveyard: "Chapel Hill Cemetery">
```

Um tweet pertence a um zumbi

```
> z = Zombie.find(2)
=> #<Zombie id: 2, name: "Bob", graveyard: "Chapel Hill Cemetery">

> t = Tweet.create(:status => "Your eyelids taste like bacon.", :zombie => z)
=> #<Tweet id: 5, status: "Your eyelids taste like bacon.", zombie_id: 2>
```


Um tweet pertence a um zumbi

```
> z = Zombie.find(2)
=> #<Zombie id: 2, name: "Bob", graveyard: "Chapel Hill Cemetery">

> t = Tweet.create(:status => "Your eyelids taste like bacon.", :zombie => z)
=> #<Tweet id: 5, status: "Your eyelids taste like bacon.", zombie_id: 2>

> t.zombie
=> #<Zombie id: 2, name: "Bob", graveyard: "Chapel Hill Cemetery">
```

Um tweet pertence a um zumbi

```
> z = Zombie.find(2)
=> #<Zombie id: 2, name: "Bob", graveyard: "Chapel Hill Cemetery">

> t = Tweet.create(:status => "Your eyelids taste like bacon.", :zombie => z)
=> #<Tweet id: 5, status: "Your eyelids taste like bacon.", zombie_id: 2>

> t.zombie
=> #<Zombie id: 2, name: "Bob", graveyard: "Chapel Hill Cemetery">

> t.zombie.name
=> "Bob"
```

Um zumbi tem
muitos tweets

```
> ash = Zombie.find(1)  
=> #<Zombie id: 1, name: "Ash", graveyard: "Glen Haven Cemetery">
```

Um zumbi tem
muitos tweets

```
> ash = Zombie.find(1)
=> #<Zombie id: 1, name: "Ash", graveyard: "Glen Haven Cemetery">

> ash.tweets.count
=> 4
```

Um zumbi tem muitos tweets

```
> ash = Zombie.find(1)
=> #<Zombie id: 1, name: "Ash", graveyard: "Glen Haven Cemetery">

> ash.tweets.count
=> 4

> ash.tweets
=> [#<Tweet id: 1, status: "Where can I get a good bite to eat?",
zombie_id: 1>, #<Tweet id: 4, status: "OMG, my fingers turned green. #FML",
zombie_id: 1>]
```