

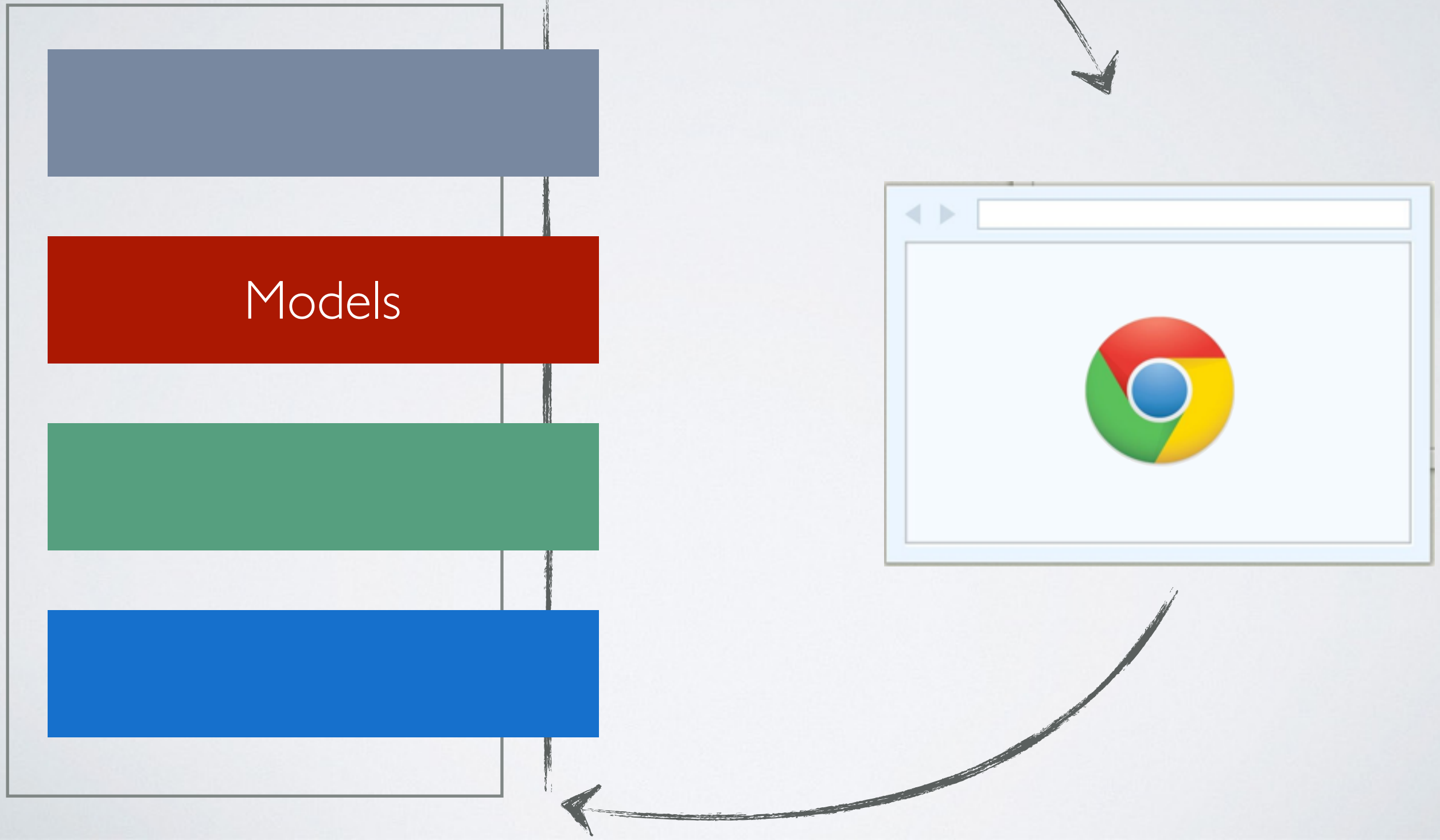
MODELOS

MODELOS

comunicación con la BD



CAPAS DE LA APLICACIÓN



t = *Tweet.find*(3)

app/model/tweet.rb

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

tweets

id	estado	monstruo
1	¡Me voy a chupar tu sangre!	Drácula
2	¡Te voy a aplastar!	Godzilla
3	¡Te voy a destripar!	King Kong
4	Feliz Primavera	Nahuelito

`t = Tweet.find(3)`

app/model/tweet.rb

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

Mapea la clase con la tabla

tweets

id	estado	monstruo
1	¡Me voy a chupar tu sangre!	Drácula
2	¡Te voy a aplastar!	Godzilla
3	¡Te voy a destripar!	King Kong
4	Feliz Primavera	Nahuelito

app/model/tweet.rb

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

Instancia de Tweet #3



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

tweets

id	estado	monstruo
1	¡Me voy a chupar tu sangre!	Drácula
2	¡Te voy a aplastar!	Godzilla
3	¡Te voy a destripar!	King Kong
4	Feliz Primavera	Nahuelito

VALIDACIONES

tweets

id	estado	monstruo
1	¡Me voy a chupar tu sangre!	Drácula
2	¡Te voy a aplastar!	Godzilla
3	¡Te voy a destripar!	King Kong
4	Feliz Primavera	Nahuelito
5		

t = *Tweet.new*

t.save



app/model/tweet.rb

```
class Tweet < ActiveRecord::Base  
  validates :estado, presence: true  
end
```

t = Tweet.new

=> #<Tweet id: nil, estado: nil, monstruo: nil>

t.save

=> false

t.errors.messages

=> {estado: ["no puede estar en blanco"]}

t.errors[:estado]

=> "no puede estar en blanco"

validates :estado,

presence: true,

uniqueness: true,

numericality: true,

length: { minimum: 0, maximum: 2000 }

format: { with: /./ }*

acceptance: true

confirmation: true

validates :estado, presence: true

validates :brazos, numericality: true

validates :huella, uniqueness: true

validates :clave, length: { minimum: 3, maximum: 20 }

validates :age, inclusion: { in: 21..99 }

validates :age, exclusion: { 0...21 }, message: “Debes ser mayor a 21”

RELACIONES



tweets

id	estado	monstruo
1	Barilo Barilo, nos vamo a Barilo	Nahuelito
2	¡Te voy a aplastar!	Godzilla
3	¡Te voy a destripar!	King Kong
4	Feliz Primavera	Nahuelito

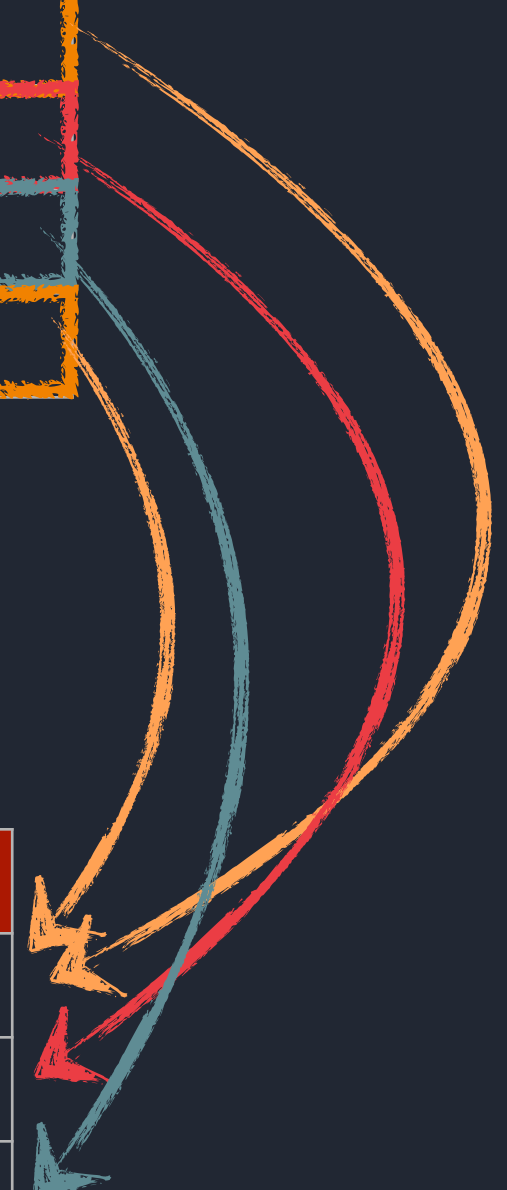
Queremos guardar los monstruos en su propia tabla

tweets

id	estado	monstruo_id
1	Barilo Barilo, nos vamo a Barilo	1
2	¡Te voy a aplastar!	2
3	¡Te voy a destripar!	3
4	Feliz Primavera	1

monstruos

id	nombre	descripcion
1	Nahuelito	Vive en el Nahuel Huapi
2	Drácula	Chupa sangre
3	King Kong	Es un gorila gigante
4	Godzilla	¿Es japonés?



RELACIONES

monstruos

id	nombre	descripcion
1	Nahuelito	Vive en el Nahuel Huapi
2	Drácula	Chupa sangre
3	King Kong	Es un gorila gigante
4	Godzilla	¿Es japonés?

un Monstruo
HAS MANY
Tweets



app/models/monstruo.rb

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

Plural



tweets

id	estado	monstruo_id
1	Barilo Barilo, nos vamo a Barilo	1
2	¡Te voy a aplastar!	2
3	¡Te voy a destripar!	3
4	Feliz Primavera	1

RELACIONES

tweets

id	estado	monstruo_id
1	Barilo Barilo, nos vamo a Barilo	1
2	¡Te voy a aplastar!	2
3	¡Te voy a destripar!	3
4	Feliz Primavera	1

un Tweet
BELONGS TO
Monstruo



app/models/tweet.rb

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

Singular



monstruos

id	nombre	descripcion
1	Nahuelito	Vive en el Nahuel Huapi
2	Drácula	Chupa sangre
3	King Kong	Es un gorila gigante
4	Godzilla	¿Es japonés?

tweets

id	estado	monstruo_id
1	Barilo Barilo, nos vamo a Barilo	1
2	¡Te voy a aplastar!	2
3	¡Te voy a destripar!	3
4	Feliz Primavera	1
5	Yendo a la Abuela Goye	1

```
t = Tweet.create(estado: "Yendo a la Abuela Goye",  
                 monstruo_id: 1)
```

USANDO RELACIONES

nahuelito = Monstruo.find(1)

=> #<Monstruo id:1, nombre: "Nahuelito", descripcion: "Vive en el Nahuel Huapi">

t = Tweet.create(estado: "Yendo a la Abuela Goye", monstruo: nahuelito)

=> #<Tweet id:5, estado: "Yendo a la Abuela Goye", monstruo_id: 1>

nahuelito.tweets.count

=> 3

nahuelito.tweets

=> [#<Tweet id: 1, estado: "Barilo Barilo, nos vamos a Barilo", monstruo_id: 1>,

#<Tweet id: 4, estado: "Feliz Primavera", monstruo_id: 1>,

#<Tweet id: 5, estado: "Yendo a la Abuela Goye", monstruo_id: 1>]

id	estado	monstruo_id
1	Barilo Barilo, nos vamo a Barilo	1
2	¡Te voy a aplastar!	2
3	¡Te voy a destripar!	3
4	Feliz Primavera	1
5	Yendo a la Abuela Goye	1

t = Tweet.find(5)

=> #<Tweet id:5, estado: "Yendo a la Abuela Goye", monstruo_id: 1>

t.monstruo

=> #<Monstruo id:1, nombre: "Nahuelito", descripcion: "Vive en el Nahuel Huapi">

t.monstruo.nombre

=> "Nahuelito"

m = Monstruo.find(1)

=> #<Monstruo id: 1, nombre: "Nahuelito", descripcion: "Vive en el Nahuel Huapi">

m.tweets

=> [#<Tweet id: 1, estado: "Barilo Barilo, nos vamos a Barilo", monstruo_id: 1>,

#<Tweet id: 4, estado: "Feliz Primavera", monstruo_id: 1>,

#<Tweet id: 5, estado: "Yendo a la Abuela Goye", monstruo_id: 1>]

m.destroy

=> true

Tweet.find(1,4,5)

=> [#<Tweet id: 1, estado: "Barilo Barilo, nos vamos a Barilo", monstruo_id: 1>,

#<Tweet id: 4, estado: "Feliz Primavera", monstruo_id: 1>,

#<Tweet id: 5, estado: "Yendo a la Abuela Goye", monstruo_id: 1>]

```
class Monstruo < ActiveRecord::Base
  has_many :tweets, dependent: :destroy
end
```

m = Monstruo.find(1)

=> #<Monstruo id:1, nombre: "Nahuelito", descripcion: "Vive en el Nahuel Huapi">

m.destroy

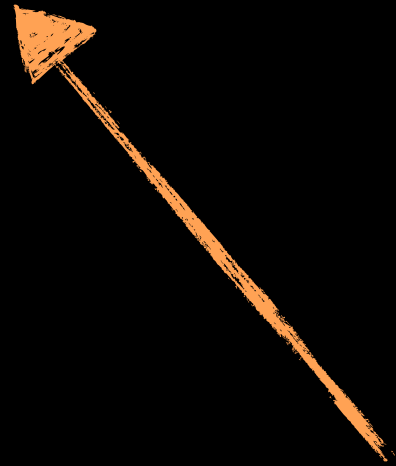
=> true

Tweet.find(1,4,5)

=> []

SCOPES

```
Tweet.order("created_at desc")
```



Lo vamos a usar muchas veces

```
class Tweet < ActiveRecord::Base
  default_scope -> { order("created_at desc") }
end
```

```
class Tweet < ActiveRecord::Base
  default_scope -> { order("created_at desc") }
end
```

Tweet.all

Retorna los tweets ordenados
por fecha descendente



`Tweet.all`

*Retorna los tweets ordenados
por fecha descendente*

`Tweet.recientes`

Últimos 3 tweets

`Tweet.eliminados`

Tweets eliminados

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

```
  default_scope -> { order("created_at desc") }
```

```
  scope :recientes, -> { order("created_at desc").limit(3) }
```

```
  scope :eliminados, -> { where("eliminado = true") }
```

```
end
```



<https://goo.gl/82Y0P8>