

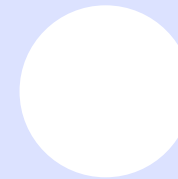
MIGRACIONES Y ASOCIACIONES

IIC2143 2022-2

¿Qué veremos?



Migraciones



Asociación 1:n



Asociación n:n



Migraciones



Migraciones

Código que realiza algún cambio en la base de datos, sin la necesidad de ejecutar una consulta de SQL



```
rails g migration AddPartNumberToProducts
```





```
class AddPartNumberToProducts < ActiveRecord::Migration[5.0]
  def change
  end
end
```



```
class AddPartNumberToProducts < ActiveRecord::Migration[5.0]
  def change
    add_column :products, :part_number, :string
  end
end
```



```
add_column
add_foreign_key
add_index
add_reference
add_timestamps
change_column_default (must supply a :from and :to option)
change_column_null
create_join_table
create_table
disable_extension
drop_join_table
drop_table (must supply a block)
enable_extension
remove_column (must supply a type)
remove_foreign_key (must supply a second table)
remove_index
remove_reference
remove_timestamps
rename_column
rename_index
rename_table
```




```
rails g migration AddDetailsToProducts part_number:string  
rails g migration RemovePartNumberFromProducts part_number:string
```





```
rails db:migrate # ejecuta todas las migraciones que no han sido ejecutadas  
rails db:rollback # deshace la última migración ejecutada
```



Asociación 1:n



movies

id	name	year	category
1	Interstellar	2014	SciFi
2	The Revenant	2015	Drama
3	Harry Potter 3	2004	Fantasy
4	Inception	2010	SciFi

directors

id	name
1	C. Nolan
2	A. Iñárratu
3	A. Cuarón

movies

id	name	year	category	director_id
1	Interstellar	2014	SciFi	1
2	The Revenant	2015	Drama	2
3	Harry Potter 3	2004	Fantasy	3
4	Inception	2010	SciFi	1



```
class Director < ApplicationRecord  
end
```

```
class Movie < ApplicationRecord  
end
```





```
director = Director.find(1)
Movie.where(director_id: director.id) # me entrega todas las películas dirigidas por el director con id 1

movie = Movie.find(1)
Director.find(movie.director_id) # me entrega el director de la película con id 1
```



```
class Director < ApplicationRecord
  has_many :movies
end
```

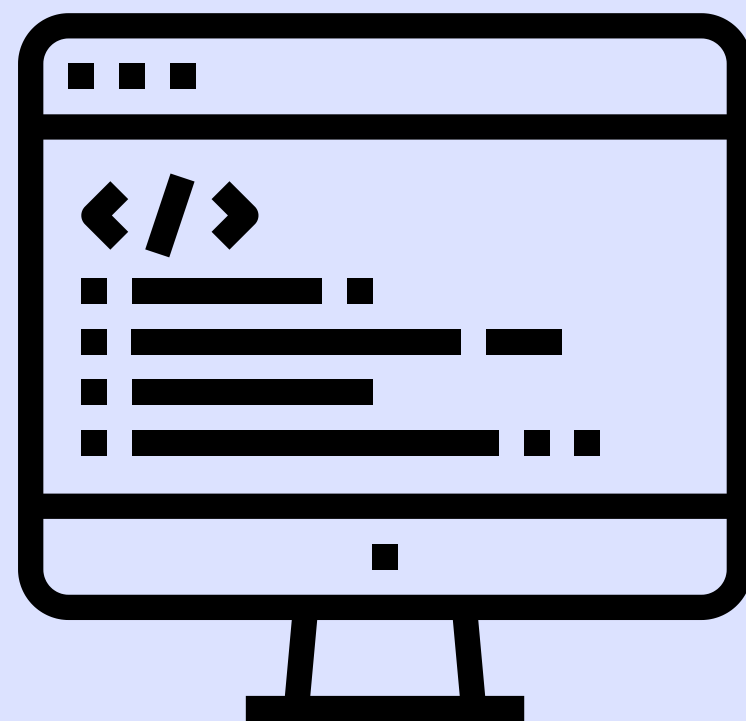
```
class Movie < ApplicationRecord
  belongs_to :director
end
```





```
director = Director.find(1)
director.movies # me entrega todas las películas dirigidas por el director con id 1

movie = Movie.find(1)
movie.director # me entrega el director de la película con id 1
```



Ejemplo



Asociación n:n



movies

id	name	year	category	director_id
1	Interstellar	2014	SciFi	1
2	The Revenant	2015	Drama	2
3	Harry Potter 3	2004	Fantasy	3
4	Inception	2010	SciFi	1

actors

id	name
1	Matthew McConaughey
2	Leonardo DiCaprio
3	Daniel Radcliffe

actors_movies

movie_id	actor_id
1	1
2	2
3	3
4	2



```
class Actor < ApplicationRecord  
  has_and_belongs_to_many :movies  
end
```

```
class Movie < ApplicationRecord  
  has_and_belongs_to_many :actors  
end
```

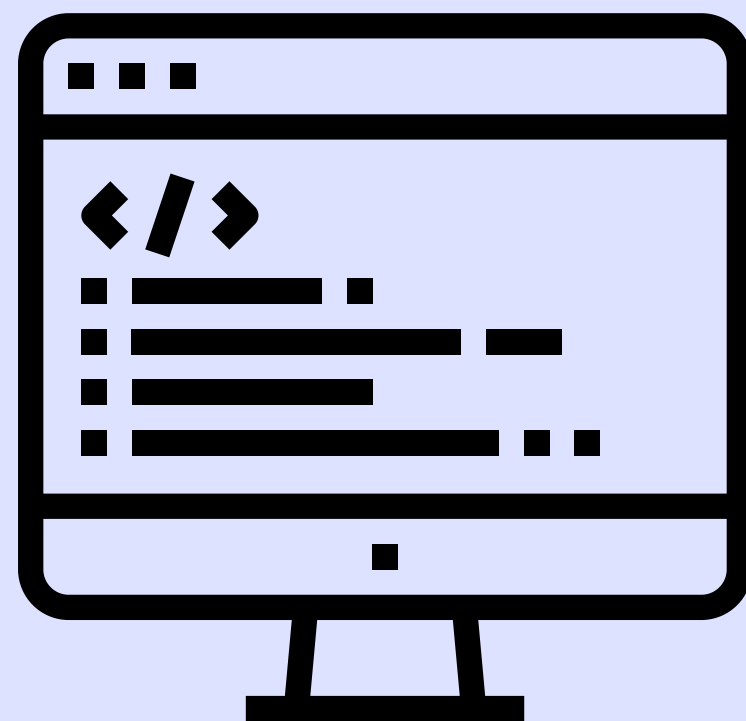


```
actor = Actor.find(1)
```

```
actor.movies # me entrega todas las películas donde actúa el actor con id 1
```

```
movie = Movie.find(1)
```

```
movie.actors # me entrega todos los actores que actúan en la película con id 1
```



Ejemplo

Conceptos interesantes

Asociación has_many :through

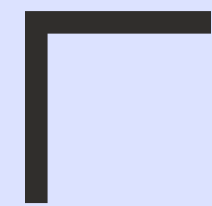
https://guides.rubyonrails.org/v6.0/association_basics.html#the-has-many-through-association

N + 1 queries problem

https://guides.rubyonrails.org/v6.0/active_record_querying.html#eager-loading-associations

Shallow nesting

<https://guides.rubyonrails.org/v6.0/routing.html#shallow-nesting>



MIGRACIONES Y ASOCIACIONES

IIC2143 2022-2