



Rails Associations & Migrations

IIC2143 - Ingeniería de Software



A decorative graphic on the left side of the slide. It features a large, light blue hexagon in the center. Surrounding it are several smaller hexagons in various shades of blue and teal. Some of these smaller hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, and a gear. There is also a network-like icon with a central node and five connecting lines. The entire graphic is set against a dark blue background.

1

Associations



¿Qué es?

Es una conexión entre dos Modelos, en este caso, de **Active Records**.

¿Por qué es necesario?

Para facilitar operaciones básica, por medio del código.





¿IIC2413 – Bases de datos?

CARRIES DEL TEAM AHORA!





Tipos de asociaciones

- ◇ belongs_to
- ◇ has_one
- ◇ has_many
- ◇ has_many :through
- ◇ has_one :through
- ◇ has_and_belongs_to_many

PRIMARY KEY - FOREIGN KEY



Ejemplos

books	
Model: Book belongs_to :author	
id	integer
author_id	integer
published_at	datetime

authors	
Model: Author	
id	integer
name	string

```
class Book < ApplicationRecord
  belongs_to :author
end
```

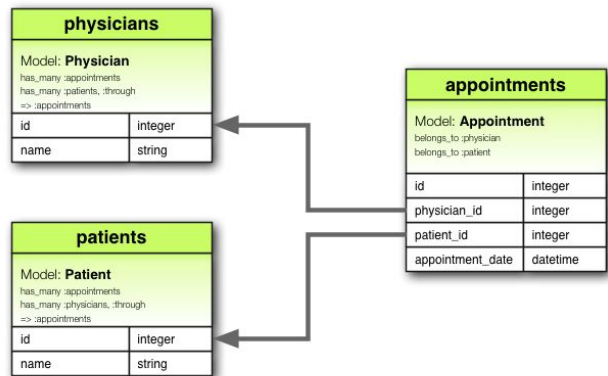


N es a N, la típica

¿Cómo representamos una
relación Many To Many?



Ejemplos



```
class Physician < ApplicationRecord
  has_many :appointments
  has_many :patients, :through => :appointments
end

class Appointment < ApplicationRecord
  belongs_to :physician
  belongs_to :patient
end

class Patient < ApplicationRecord
  has_many :appointments
  has_many :physicians, :through => :appointments
end
```


A decorative graphic on the left side of the slide. It features a large cyan hexagon in the center containing the number '2'. Surrounding this central hexagon are several smaller hexagons of varying shades of blue and cyan. Some of these smaller hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, and a gear. There is also a network-like icon with a central node and several smaller nodes connected by lines. The entire graphic is set against a dark blue background.

2

Migrations



¿Qué es?

Forma de alterar el esquema de la base de datos en el tiempo, de una manera consistente y fácil.

¿Por qué es necesario?

Para interactuar con la base de datos por medio de código, dando instrucciones.





¿Para qué nos sirve?

- ◇ **Alterar nuestra base de datos**
- ◇ **Mantener distintos modelos en cada rama de trabajo**
- ◇ **Desligarnos del trabajo de la base de datos**





Ejemplos

```
$ rails generate migration AddPartNumberToProducts  
part_number:string:index
```

```
$ rails generate migration RemovePartNumberFromProducts  
part_number:string
```

```
$ rails generate migration AddUserRefToProducts user:references
```



¿Cómo utilizarlas?

- ◇ Docker-compose run web ...
- ◇ generate = g
- ◇ Las migraciones se crean, y luego se MIGRAN (db:migrate)





¿Qué más podemos hacer?

- ◇ Cambiar nombre de atributo
- ◇ Agregar valores por defecto
- ◇ Cambiar valores por defecto
- ◇ Crear tablas
- ◇ Agregar columnas
- ◇ etc.



A decorative graphic on the left side of the slide. It features a large cyan hexagon in the center with the number '3' inside. Surrounding this central hexagon are several smaller hexagons of varying shades of blue and cyan. Some of these smaller hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, and a gear. There is also a network-like icon with a central node and several smaller nodes connected by lines.

3

Errores TÍPICOS



Orden de Ejecución

Hay que asegurarse bien qué entidad va antes. No podemos instanciar referencias a entidades que no existen.



```
• class CreateAlbums < ActiveRecord::Migration[5.1]
  def change
    create_table :albums do |t|
      t.string :name
      t.integer :year
      t.integer :total_duration
      t.references :artist, foreign_key: true
      t.timestamps
    end
  end
end
```

```
• class CreateArtists < ActiveRecord::Migration[5.1]
  def change
    create_table :artists do |t|
      t.string :name
      t.integer :birth_year
      t.integer :first_album_year
      t.timestamps
    end
  end
end
```



Editar Migraciones Anteriores

Cada migración es única, si desean hacer un cambio a la migración, CREEN OTRA!



```
• class CreateArtists < ActiveRecord::Migration[5.1]
  def change
    create_table :artists do |t|
      t.string :name
      t.integer :birth_year
      t.integer :first_album_year
      t.string :this_is_a_cool_new_attribute
      t.timestamps
    end
  end
end
```





Valores por Defecto

Los valores se setean por defecto en **nil**, ojo con los valores cuando no se agregan.



```
• class CreateAlbums < ActiveRecord::Migration[5.1]
  def change
    create_table :albums do |t|
      t.string :name
      t.integer :year
      t.integer :total_duration, default: 0
      t.references :artist, foreign_key: true
      t.timestamps
    end
  end
end
```



TIPS





Editar migración

Antes de hacer **db:migrate**, editen todo lo que necesiten en la migración.



Atributos Obligatorios

Si tienen una entidad con atributos obligatorios, **SETEEN POR DEFECTO** el valor de ese atributo.



Ya migré y la embarré, ¿que hago?

Crean una nueva migración que les ayude a deshacer o hacer lo que les faltó



Quiero retroceder en una migración

Este caso extremo se puede llevar a cabo con:
`docker-compose run web rake db:rollback`



Referencias

- ◇ https://guides.rubyonrails.org/v5.2/association_basics.html
- ◇ https://edgeguides.rubyonrails.org/active_record_migrations.html

