

Rails Migrations & Associations

IIC2143 Ingeniería de Software - Ayudantía 5

Contenidos

1. *Migrations*

- ❖ Tipos de *Migrations*
- ❖ Recomendaciones

2. *Associations*

- ❖ Tipos de *Associations*
- ❖ Ejemplos

3. Referencias

Migrations

¿Qué es una migración?

- ❖ Cambio en la base de datos (?)
- ❖ Reescribe la base de datos indirectamente (?)
- ❖ Genera un registro de los cambios en la base de datos (?)
- ❖ Cambia las *Associations* dentro del modelo (?)

¿Qué es una migración?

- ❖ Cambio en la base de datos

Si, permite realizar cambios en la base de datos

- ❖ Reescribe la base de datos indirectamente

Si, así no tienes que interactuar con el sistema gestor (PostgreSQL)

- ❖ Genera un registro de los cambios en la base de datos

Si, en la carpeta db/migrate estan las migraciones realizadas. Además, esto permite gitflow (?)

- ❖ Cambia las *Associations* dentro del modelo

No, si realizaste un cambio en las asociaciones, debes generar una migración y cambiar lo escrito dentro del modelo.

Tipos de migración: Standalone

`docker-compose run web ...`

- ❏ empty migration

`rails g migration NameOfMigration`

- ❏ add_column

`rails g migration AddColumnToTable column:type`

- ❏ remove_column

`rails g migration RemoveColumnsFromTable
column_1:type column_2:type`

- ❏ add_reference (remove_reference)

`rails g migration AddModelNameToTable
model_name:references`

Tipo de migración: Model (Scaffold)

```
rails g model ModelName attribute:type attribute2:type  
rails g scaffold ModelName attribute:type attribute2:type
```

Escribir una empty migration

```
class MigrationFoo < ActiveRecord::Migration[5.1]
  def change
    add_column :products, :price, :decimal, precision: 5, scale: 2
    add_reference :products, :supplier, polymorphic: true
    rename_column :table, :old_column, :new_column

    create_table :products do |t|
      t.string :name
      t.text :description

      t.timestamps
    end
  end
end
```


Recomendaciones

- ❖ Cuidar el orden de ejecución
- ❖ Hacer los cambios necesarios antes de migrar
- ❖ Si son atributos obligatorios, agregar un default
- ❖ Si es necesario deshacer cambios de una migración, realizar una migración para eso y NO editar migraciones anteriores

Associations

¿Qué es una asociación?

- ❏ Conexión entre modelos (?)
- ❏ Interacción entre modelos (?)
- ❏ Modela Composición y Agregación (?)

¿Qué es una asociación?

- ❖ **Conexión entre modelos**

Si, permite relacionar los modelos entre ellos

- ❖ **Interacción entre modelos**

Si y no, pues las interacciones se manejan a nivel Controlador, y la asociación es a nivel Modelo.

- ❖ **Modela Composición y Agregación**

Si, permite asociar modelos de manera opcional en el caso de agregación

Tipos de asociación

- `belongs_to`
- `has_one`
- `has_many`
- `has_many :through`
- `has_one :through`
- `has_and_belongs_to_many`

belongs_to & has_many

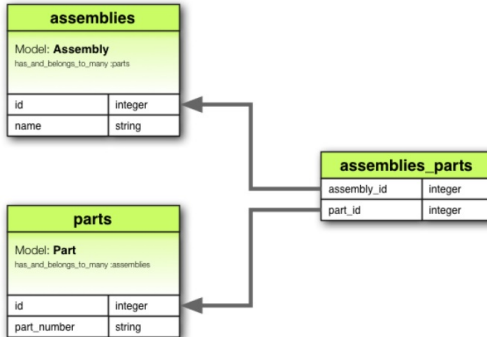


```
class Book < ApplicationRecord
  belongs_to :author, optional: true
end

class Author < ApplicationRecord
  has_many :books
end
```

Para relaciones 1 - 1, sólo se cambia el `has_many` por un `has_one`
En caso de ser agregación, utilizar la opción `optional: true`

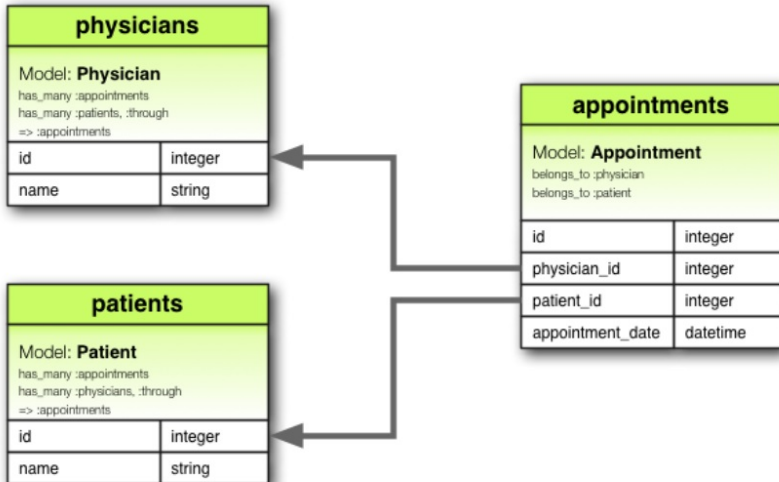
has_and_belongs_to_many



```
class Assembly < ApplicationRecord
  has_and_belongs_to_many :parts
end

class Part < ApplicationRecord
  has_and_belongs_to_many :assemblies
end
```

has_many :through



Referencias

❏ Migrations

- https://edgeguides.rubyonrails.org/active_record_migrations.html

❏ Associations

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