



Relaciones 1 a N en los modelos

Asociando elementos

***Implementar relaciones
1 a N en los modelos
utilizando Ruby on Rails
para satisfacer un
requerimiento.***

- Unidad 1: Relaciones 1 a N.
- Unidad 2: Mecanismos de autenticación y control de accesos en una aplicación web.
- Unidad 3: Relaciones N a N en los modelos.
- Unidad 4: Utilizar Amazon S3 en proyecto en Rails.
- Unidad 5: Pruebas unitarias y funcionales.



Te encuentras aquí



¿Qué aprenderás en esta sesión?

- *Asociar elementos con inputs avanzados y rutas anidadas.*

¿Qué input debe ser
agregado al formulario
para poder aceptar
campos de otro modelo?



¿Qué pasa si agregamos
el método `build` en el
método de controlador
`create`?



¿Se puede crear un
formulario anidado sin usar
el método build?



/* Asociando elementos */

Asociando elementos

Select

El input select en un formulario despliega una lista para poder realizar una selección del elemento deseado.

Radio

El input radio es utilizado para poder seleccionar o chequear más de un propietario a la vez.

Ejercicio guiado

"Librería"



Librería

Necesitamos ordenar los libros que se encuentran en la librería por su autor y por su editorial, para poder lograr esto sabemos lo siguiente:

- Un libro no puede existir si no está relacionado con un autor.
- Un libro solo puede ser publicado por una editorial.
- Un libro puede tener muchos autores.



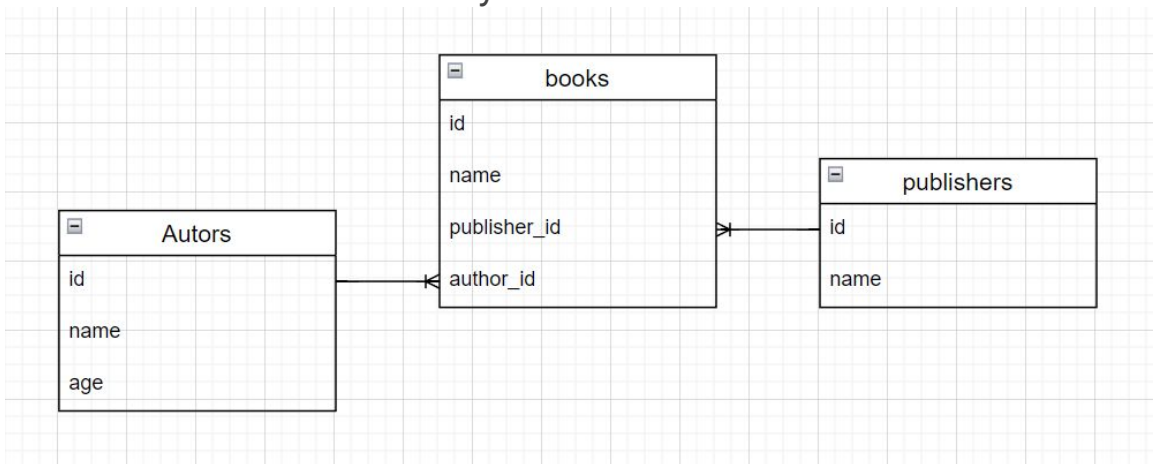
Librería

Solución

- **Paso 1:** Creamos el proyecto con base de datos PostgreSQL.

```
#bashrc  
rails new bookstore -d postgresql
```

- **Paso 2:** Identificamos las entidades y sus relaciones.



Librería

Solución

- **Paso 3:** Creamos la base de datos.

```
#bashrc  
rails db:create
```

- **Paso 4:** Creamos scaffold para libros, autores y editorial.

```
#bashrc  
rails g scaffold autor name age:integer  
  
rails g scaffold publisher name  
  
rails g scaffold book name
```



Librería

Solución

- **Paso 5:** Agregamos claves foráneas a modelo books

```
#bashrc
```

```
rails g migration AddDetailsToBooks publisher:references autor:references
```

```
class AddDetailsToBooks < ActiveRecord::Migration[7.0]
  def change
    add_reference :books, :publisher, null: false, foreign_key: true
    add_reference :books, :autor, null: false, foreign_key: true
  end
end
```

Librería

Solución

- **Paso 6:** Relacionamos los modelos.

```
#app/models/autor.rb
class Autor < ApplicationRecord
  has_many :books
end

#app/models/publisher.rb
class Publisher < ApplicationRecord
  has_many :books
end

#app/models/book.rb
class Book < ApplicationRecord
  belongs_to :publisher
  belongs_to :autor
end
```



Librería

Solución

- **Paso 7:** Creamos mediante Rails console, autores y editoriales.

```
#rails c
Autor.create!(name: "JK RawrLin")
Autor.create!(name: "Jorge R. R. Martin")
Autor.create!(name: "Gabo Marquez")
Autor.create!(name: "Neftali Relles")
Autor.create!(name: "Gaby miztral")
Autor.create!(name: "Stifen King")
Autor.create!(name: "Carlos Nono")
Autor.create!(name: "HP LoveMake")

Publisher.create!(name: "Arcano III")
Publisher.create!(name: "Ediciones SM")
Publisher.create!(name: "Tres Vientos")
```



Librería

Solución

- **Paso 8:** Anidamos las rutas de autores y libros.

```
#config/routes.rb
Rails.application.routes.draw do
  resources :publishers
  resources :authors do
    resources :books
  end
end
```

autor_books	GET	/authors/:autor_id/books(.:format)	books#index
	POST	/authors/:autor_id/books(.:format)	books#create
new_autor_book	GET	/authors/:autor_id/books/new(.:format)	books#new
edit_autor_book	GET	/authors/:autor_id/books/:id/edit(.:format)	books#edit
autor_book	GET	/authors/:autor_id/books/:id(.:format)	books#show
	PATCH	/authors/:autor_id/books/:id(.:format)	books#update
	PUT	/authors/:autor_id/books/:id(.:format)	books#update
	DELETE	/authors/:autor_id/books/:id(.:format)	books#destroy

Librería

Solución

- **Paso 9:** Modificamos los métodos del controlador libros según las rutas.

Antes de modificar las rutas vamos a crear un método auxiliar para evitar repetir código innecesariamente

```
#app/controllers/books_controller.rb
before_action :set_autor

def set_autor
  @autor = Autor.find(params[:autor_id])
end
```



Librería

Solución

- **Paso 9:** Modificamos los métodos del controlador libros según las rutas.
 - **9.1:** Modificar método index de libros

```
autor_books GET    /authors/:autor_id/books(.:format)  
books#index
```

```
#app/controllers/books_controller.rb  
def index  
  @books = Book.all  
end
```



```
#app/controllers/books_controller.rb  
def index  
  #@autor = Autor.find(params[:autor_id])  
  @books = @autor.books  
end
```

Librería

Solución

- **Paso 9:** Modificamos los métodos del controlador libros según las rutas.
 - **9.2:** Método new

```
new_autor_book GET    /autors/:autor_id/books/new(.:format)  books#new
```

```
#app/controllers/books_controller.rb  
def new  
  @book = Book.new  
end
```



```
#app/controllers/books_controller.rb  
def new  
  @book = Book.new  
end
```

Librería

Solución

- **Paso 9:** Modificamos los métodos del controlador libros según las rutas.
 - **9.3:** Método create

POST /authors/:autor_id/books(.:format)

books#create

```
#app/controllers/books_controller.rb
def create
  @book = Book.new(book_params)

  respond_to do |format|
    if @book.save
      format.html { redirect_to book_url(@book), notice: "Book was
successfully created." }
      format.json { render :show, status: :created, location: @book }
    else
      format.html { render :new, status: :unprocessable_entity }
      format.json { render json: @book.errors, status:
:unprocessable_entity }
    end
  end
end
```



```
#app/controllers/books_controller.rb
def create
  @book = Book.new(book_params)
  respond_to do |format|
    if @book.save!
      format.html { redirect_to autor_book_path(@autor.id, @book.id), notice: "Book
was successfully created." }
      format.json { render :show, status: :created, location: @book }
    else
      format.html { render :new, status: :unprocessable_entity }
      format.json { render json: @book.errors, status: :unprocessable_entity }
    end
  end
end
```

Librería

Solución

- **Paso 9:** Modificamos los métodos del controlador libros según las rutas.
 - **9.4:** Método edit

```
edit_autor_book GET /autors/:autor_id/books/:id/edit(.:format)
booksks#edit
```

```
#app/controllers/books_controller.rb
def edit
end
```



```
#app/controllers/books_controller.rb
def edit
end
```



Librería

Solución

- **Paso 9:** Modificamos los métodos del controlador libros según las rutas.
 - **9.5:** Método update

PATCH /authors/:autor_id/books/:id(.:format) books#update

```
#app/controllers/books_controller.rb
def update
  respond_to do |format|
    if @book.update(book_params)
      format.html { redirect_to book_url(@book), notice: "Book was
successfully updated." }
      format.json { render :show, status: :ok, location: @book }
    else
      format.html { render :edit, status: :unprocessable_entity }
      format.json { render json: @book.errors, status:
:unprocessable_entity }
    end
  end
end
```



```
#app/controllers/books_controller.rb
def update
  respond_to do |format|
    if @book.update(book_params)
      format.html { redirect_to autor_book_path(@autor.id, @book.id), notice:
"Book was successfully updated." }
      format.json { render :show, status: :ok, location: @book }
    else
      format.html { render :edit, status: :unprocessable_entity }
      format.json { render json: @book.errors, status: :unprocessable_entity }
    end
  end
end
```

Librería

Solución

- **Paso 9:** Modificamos los métodos del controlador libros según las rutas.
 - **9.6:** Método show

```
autor_book GET    /authors/:autor_id/books/:id(.:format)    books#show
```

```
#app/controllers/books_controller.rb  
def show  
end
```



```
#app/controllers/books_controller.rb  
def show  
end
```

Librería

Solución

- **Paso 9:** Modificamos los métodos del controlador libros según las rutas.
 - **9.7:** Método destroy

```
DELETE /authors/:autor_id/books/:id(.:format) books#destroy
```

```
#app/controllers/books_controller.rb
def destroy
  @book.destroy

  respond_to do |format|
    format.html { redirect_to books_url, notice: "Book was successfully
destroyed." }
    format.json { head :no_content }
  end
end
```



```
#app/controllers/books_controller.rb
def destroy
  @book.destroy

  respond_to do |format|
    format.html { redirect_to autor_books_path(@autor.id), notice: "Book
was successfully destroyed." }
    format.json { head :no_content }
  end
end
```



Librería

Solución

- **Paso 9:** Modificamos los métodos del controlador libros según las rutas. Por último vamos a editar los strongs params para tener el controlador listo, con esta solución y lo que sabemos hasta ahora, nos vamos a quedar cortos dado que un libro no podrá tener muchos autores, pero en un futuro cuando tengamos más conocimientos volveremos al proyecto a terminar sus últimos detalles.

```
#app/controllers/books_controller.rb
def book_params
  params.require(:book).permit(:name, :autor_id, :publisher_id)
end
```



Librería

Solución

- **Paso 10:** Modificamos las rutas en las vistas.
 - **10.1:** Vista index

```
#app/views/books/index.html.erb
<p style="color: green"><%= notice %></p>
<h1>Books</h1>
<div id="books">
  <% @books.each do |book| %>
    <%= render book %>
    <p>
      <%= link_to "Show this book", autor_book_path(@autor.id, book.id) %>
    </p>
  <% end %>
</div>
<%= link_to "New book", new_autor_book_path(@autor.id) %>
```



Librería

Solución

- **Paso 10:** Modificamos las rutas en las vistas.
 - **10.2:** Vista new

```
#app/views/books/new.html.erb
<h1>New book</h1>
<%= render "form", book: @book %>
<br>
<div>
  <%= link_to "Back to books", autor_books_path(@autor.id) %>
</div>
```



Librería

Solución

- **Paso 10:** Modificamos las rutas en las vistas.
 - **10.3:** Vista edit

```
#app/views/books/edit.html.erb
<h1>Editing book</h1>
<%= render "form", book: @book %>
<br>
<div>
  <%= link_to "Show this book", autor_book_path(@author.id, @book.id) %> |
  <%= link_to "Back to books", autor_books_path(@autor.id) %>
</div>
```



Librería

Solución

- **Paso 10:** Modificamos las rutas en las vistas.
 - **10.4:** Vista show

```
#app/views/books/show.html.erb
<p style="color: green"><%= notice %></p>
<%= render @book %>
<div>
  <%= link_to "Edit this book", edit_autor_book_path(@autor.id, @book.id) %> |
  <%= link_to "Back to books", autor_books_path(@autor.id) %>
  <%= button_to "Destroy this book", autor_book_path(@autor.id, @book.id),
method: :delete %>
</div>
```



Librería

Solución

- **Paso 11:** Modificamos el formulario de libros.

```
#app/views/books/_form.html.erb
<%= form_with(model: [@autor, book]) do |form| %>
  <% if book.errors.any? %>
    <div style="color: red">
      <h2><%= pluralize(book.errors.count, "error") %> prohibited this book from being saved:</h2>
      <ul>
        <% book.errors.each do |error| %>
          <li><%= error.full_message %></li>
        <% end %>
      </ul>
    </div>
  <% end %>
  <div>
    <%= form.label :name, style: "display: block" %>
    <%= form.text_field :name %>
  </div>
  <div>
    <%= form.submit %>
  </div>
<% end %>
```



Librería

Solución

- **Paso 12:** Podemos elegir un Autor.

```
#app/views/books/_form.html.erb
<div>
  <%= form.label :author_ids, style: "display: block" %>
  <%= form.collection_radio_buttons(:author_id, Autor.all, :id, :name)%>
</div>
```

New book

Name

Author

☐ JK RawrLin ☐ Jorge R. R. Martin ☐ Gabo Marquez ☐ Gaby miztral ☐ Stifen King ☐ Carlos Nono ☐ HP LoveMake

Create Book

[Back to books](#)



Librería

Solución

- **Paso 13:** Seleccionamos una editorial.

```
#app/views/books/_form.html.erb
<div>
  <%= form.label :publisher_id, style: "display: block" %>
  <%= form.select :publisher_id, options_for_select(Publisher.all.pluck(:name, :id)) %>
</div>
```

New book

Name

Author

☐ JK RawrLin ☐ Jorge R. R. Martin ☐ Gabo Marquez ☐ Gaby miztral ☐ Stifen King ☐ Carlos Nono ☐ HP LoveMake

Publisher

Arcano III ▾

Arcano III
Ediciones SM
Tres Vientos
[Back to books](#)

¿Qué comando en la terminal
nos dice qué estructura
deben seguir las rutas?





Próxima sesión...

- *Guía de ejercicios.*

{desafío}
latam_

*Academia de
talentos digitales*

