

---

---

# Ruby on Rails

— Pagination —

---

---

# Paginación

1. Incluir la gema: **will\_paginate**
2. Run **\$bundle**
3. Incluir **una** línea de código en el controller
4. Incluir **una** línea de código en el view.
5. **Reiniciar** el servidor

# Controller

```
class BooksController < ApplicationController
  before_action :set_book, only: [:show, :edit, :update, :destroy]

  # GET /books
  # GET /books.json
  def index
    @books = current_user.books.paginat(page: params[:page], per_page: 10)
  end

  # GET /books/1
  # GET /books/1.json
  def show
  end

  # GET /books/new
  def new
    @book = current_user.books.new
  end
end
```

# View - index.html.erb

```
<tbody>
  <%= @books.each do |book| %>
    <tr>
      <td><%= book.name %></td>
      <td><%= book.author %></td>
      <td><%= link_to 'Show', book %></td>
      <td><%= link_to 'Edit', edit_book_path(book) %></td>
      <td><%= link_to 'Destroy', book, method: :delete, data: { confirm: 'Are you sure?' } %></td>
    </tr>
  <%= end %>
</tbody>
</table>
```

```
<div class="bs-component">
  <%= will_paginate @books%>
</div>
```

```
<br>
```

# Listing Books

Name Author

Book57Author57[Show](#)[Edit](#)[Destroy](#)

Book61Author61[Show](#)[Edit](#)[Destroy](#)

Book64Author64[Show](#)[Edit](#)[Destroy](#)

Book66Author66[Show](#)[Edit](#)[Destroy](#)

Book67Author67[Show](#)[Edit](#)[Destroy](#)

Book68Author68[Show](#)[Edit](#)[Destroy](#)

Book72Author72[Show](#)[Edit](#)[Destroy](#)

Book78Author78[Show](#)[Edit](#)[Destroy](#)

Book80Author80[Show](#)[Edit](#)[Destroy](#)

Book81Author81[Show](#)[Edit](#)[Destroy](#)

← Previous 1 2 3 4 5 Next →

[New Book](#)



This repository

Search

Pull requests

Issues

Gist



 bootstrap-ruby / will\_paginate-bootstrap

Watch

14

★ Star

363

Fork

92

Code

Issues 8

Pull requests 6

Projects 0

Wiki

Pulse

Graphs

Integrates the Twitter Bootstrap pagination component with will\_paginate

63 commits

2 branches

10 releases

12 contributors

MIT

Branch: master

New pull request

Create new file




Upload files

Find file

Clone or download

 **nickpad** committed on **GitHub** Add no longer maintained notice to README

Latest commit 1a0d7b3 19 days ago

 example	Remove support for Bootstrap 2.x	3 years ago
 lib	Supposed fix	11 months ago
 spec	Fix spelling in spec description	2 years ago
 .gitignore	tweak .gitignore	5 years ago
 Gemfile	add an example sinatra app	4 years ago

# Se elige el renderer

```
<tbody>
  <% @books.each do |book| %>
    <tr>
      <td><%= book.name %></td>
      <td><%= book.author %></td>
      <td><%= link_to 'Show', book %></td>
      <td><%= link_to 'Edit', edit_book_path(book) %></td>
      <td><%= link_to 'Destroy', book, method: :delete, data: { confirm:
      td>
    </tr>
  <% end %>
</tbody>
</table>
```

```
<div class="bs-component">
  <%= will_paginate @books %>
  <%= will_paginate @books, renderer: BootstrapPagination::Rails %>
</div>
```

# Listing Books

Name	Author	
Metaprogramming Ruby 2	Paolo Perrotta	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book1	Author1	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book5	Author5	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book6	Author6	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book9	Author9	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book10	Author10	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book12	Author12	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book13	Author13	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book17	Author17	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>
Book19	Author19	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>

[← Previous](#)[1](#)[2](#)[3](#)[4](#)[5](#)[Next →](#)[New Book](#)



# Entonces...

- **will\_paginate** facilita la lógica para la paginación de resultados.
- Se pueden encontrar gemas que ya aplican [bootstrap](#).
- Todo lo que necesitamos para utilizar esto es **una** línea de scoping en el controlador y **un** parámetro :page desde la vista.