

Laboratorio de Computación IV

Clase 21

Andrés Fortier

Repaso

- rspec-rails y tipos de tests

Tipo de test	Unidad	Integración
Model - No ActiveRecord	Si	No
Model - ActiveRecord	Depende	Depende
Controller	Depende	Depende
View	Depende	Depende
Routes	Depende	Depende
Helpers	Depende	Depende
Requests	No	Si
Features	No	Si

Listados



- Al mostrar listados (potencialmente grandes) surgen tres problemas recurrentes
 - Restringir los resultados (búsquedas o filtros).
 - Ordenar.
 - Pagar.

Paginación



- Colocar los índices correctos en el query
 - Ej. ``Articles.all.limit(5).offset(10)``
- Mostrar en el listado un paginador
 - Links a primero, último, anterior, siguiente, etc.
- En Rails, resuelto por gemas hace tiempo
 - Kaminary
 - Will paginate

Will paginate

```
/Gemfile
```

```
..  
gem 'will_paginate', '~> 3.1.0'  
...
```

```
$ bundle install
```

Will paginate

```
/app/controllers/articles_controller.rb
class ArticlesController < ApplicationController
  ...
  def index
    authorize Article
    @articles = Article
      .paginate(:page => params[:page], :per_page => 5)
      .order(created_at: :desc)
  end
  ...
end
```

- Vayan al índice
 - Nos falta la parte visual

Will paginate

```
/app/views/articles/index.html.erb
```

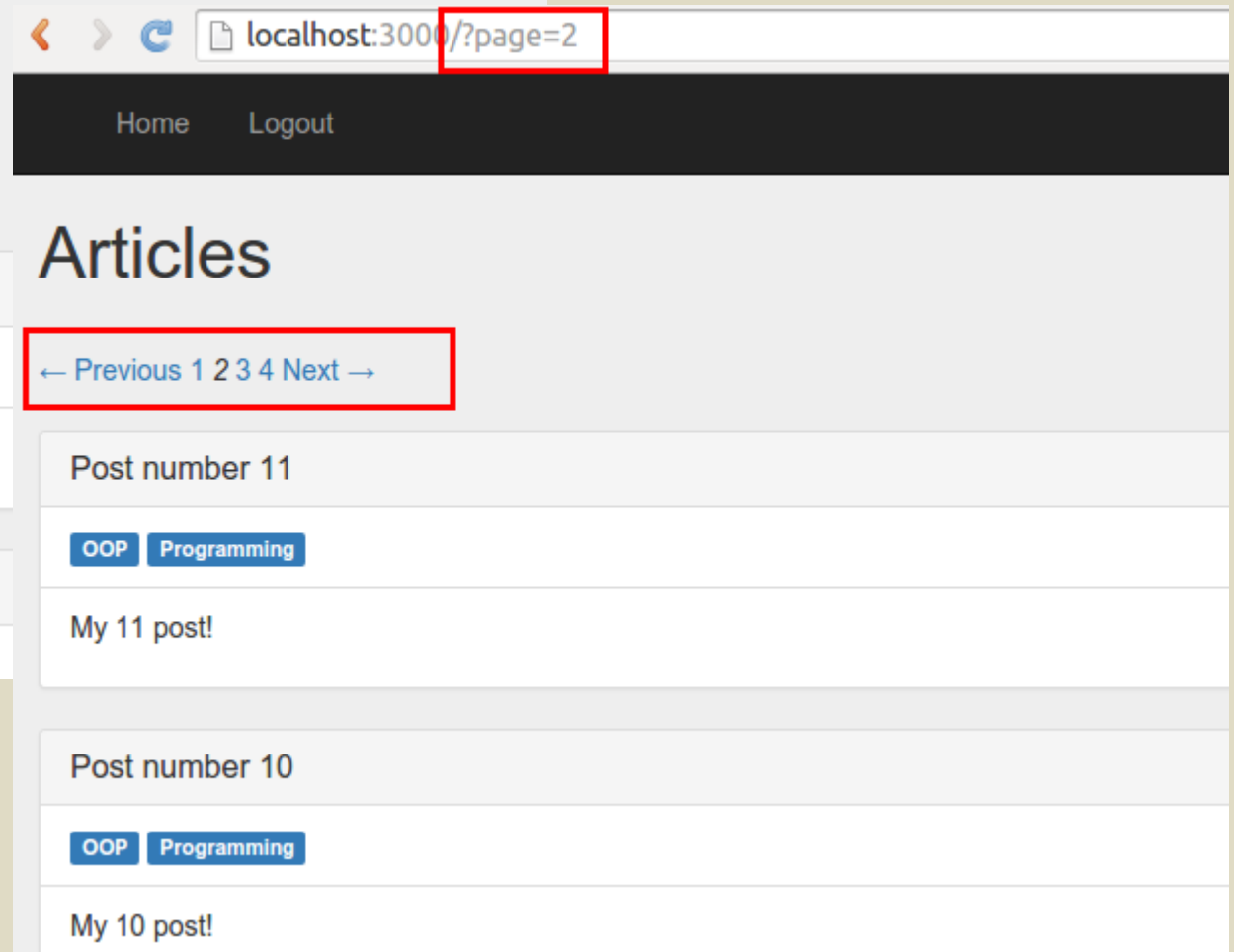
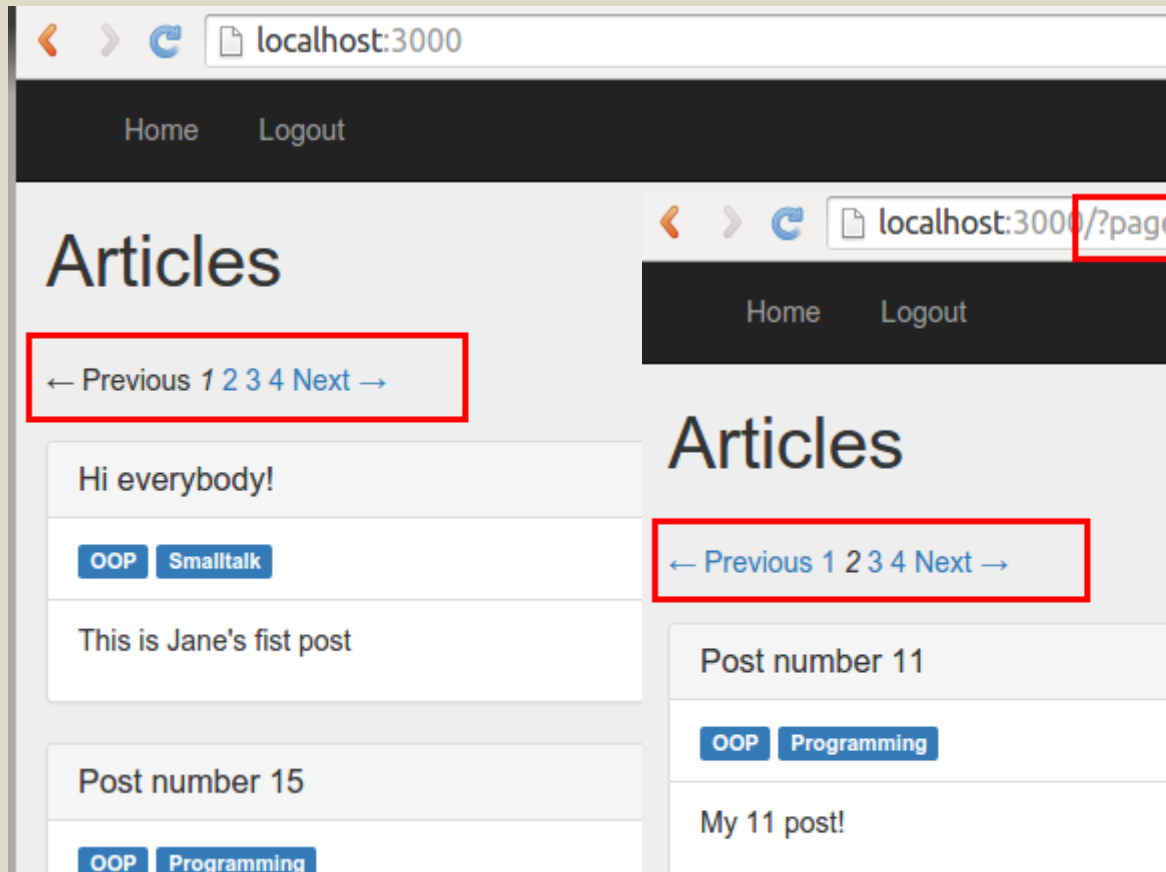
```
<h1>Articles</h1>
```

```
<%= will_paginate @articles %>
```

```
<% @articles.each do |article| %>
```

```
...
```

Will paginate



Will paginate - bootstrap

- Podemos acomodar el estilo a mano o usar una gema

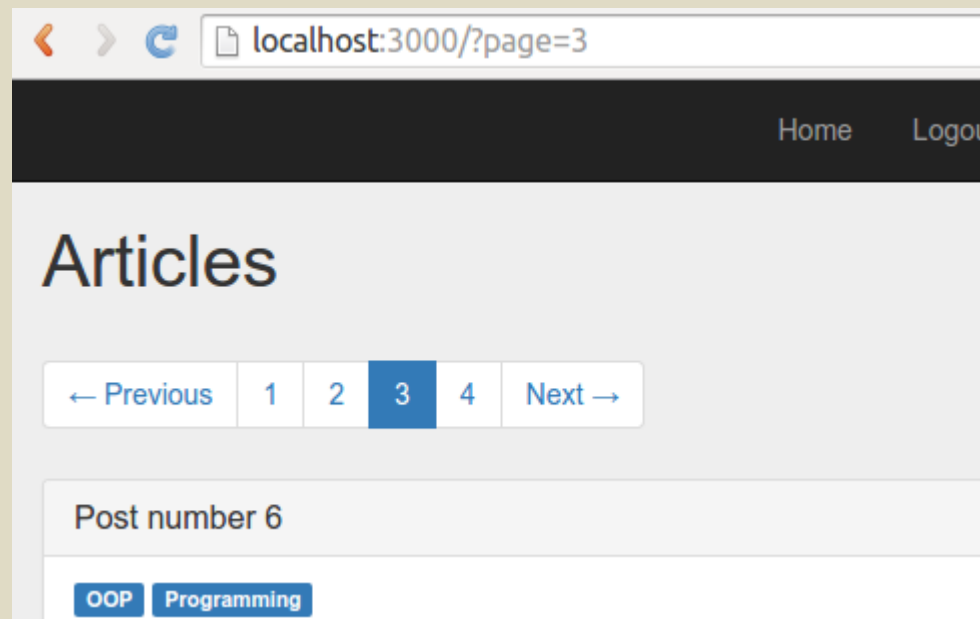
```
/Gemfile
```

```
..  
gem 'will_paginate-bootstrap', '~> 1.0.1'  
...
```

```
$ bundle install
```

Will paginate

```
/app/views/articles/index.html.erb  
<h1>Articles</h1>  
  
<%= will_paginate @articles, renderer:  
  BootstrapPagination::Rails %>  
  
<% @articles.each do |article| %>  
...  
</%>
```



Búsquedas

- Formulario para ingresar los parámetros de la búsqueda.
- Aplicar el filtro al listado de artículos.

Búsquedas

```
/app/views/articles/index.html.erb
```

```
<h1>Articles</h1>
```

```
<%= form_tag articles_path, :method => 'get' do %>
```

```
  <p>
```

```
    <%= text_field_tag :search, params[:search] %>
```

```
    <%= submit_tag "Search", :name => nil %>
```

```
  </p>
```

```
<% end %>
```

```
<%= will_paginate @articles, renderer:
```

```
BootstrapPagination::Rails %>
```

Búsquedas

```
/app/controllers/articles_controller.rb
```

```
class ArticlesController < ApplicationController
  ...
  def index
    authorize Article
    @articles = Article
      .search(params[:search])
      .paginate(:page => params[:page], :per_page => 5)
      .order(created_at: :desc)
  end
  ...
end
```

Búsquedas

```
/app/models/article.rb  
class Article < ActiveRecord::Base  
  ...  
  
  def self.search(search)  
    if search.present?  
      where({title: search})  
    else  
      all  
    end  
  end  
end  
end
```

Búsquedas

Articles

Post number 15

OOP

Programming

My 15 post!

[New article](#)

Búsquedas

- Sólo funciona para
 - Un único campo.
 - Búsquedas exactas.



Búsquedas

- Búsquedas con datos parciales

```
/app/models/article.rb
```

```
class Article < ActiveRecord::Base
  ...

  def self.search(search)
    if search.present?
      where('title LIKE ?', "%#{search}%")
    else
      all
    end
  end
end

end
```

Búsquedas

Articles

[< Previous](#)[1](#)[2](#)[3](#)[4](#)[Next >](#)

Post number 15

[OOP](#)[Programming](#)

My 15 post!

Post number 14

[OOP](#)[Programming](#)

My 14 post!

Búsquedas

- Búsquedas por mas de un atributo
 - Múltiples campos de entrada
 - Un campo, búsqueda en múltiples columnas

```
/app/models/article.rb
```

```
class Article < ActiveRecord::Base
  ...
  def self.search(search)
    if search.present?
      where('title LIKE :search OR text LIKE :search',
search: "%#{search}%")
    else
      all
    end
  end
end
...
```

Búsquedas

- Debemos aplicar bootstrap al formulario de búsqueda

```
/app/views/articles/index.html.erb
```

```
<h1>Articles</h1>
```

```
<%= form_tag articles_path, :method => 'get', class:  
'form-inline' do %>
```

```
  <div class="form-group">
```

```
    <%= text_field_tag :search, params[:search], class:  
'form-control' %>
```

```
    <%= submit_tag "Search", :name => nil, class: 'btn  
btn-primary' %>
```

```
  </div>
```

```
<% end %>
```

```
...
```

Búsquedas

Articles

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

Hi everybody!

[OOP](#)[Smalltalk](#)

This is Jane's fist post

Post number 15