

Laboratorio de Computación IV

Clase 20

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', '~> 3.0.7'  
...
```

```
$ 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
```

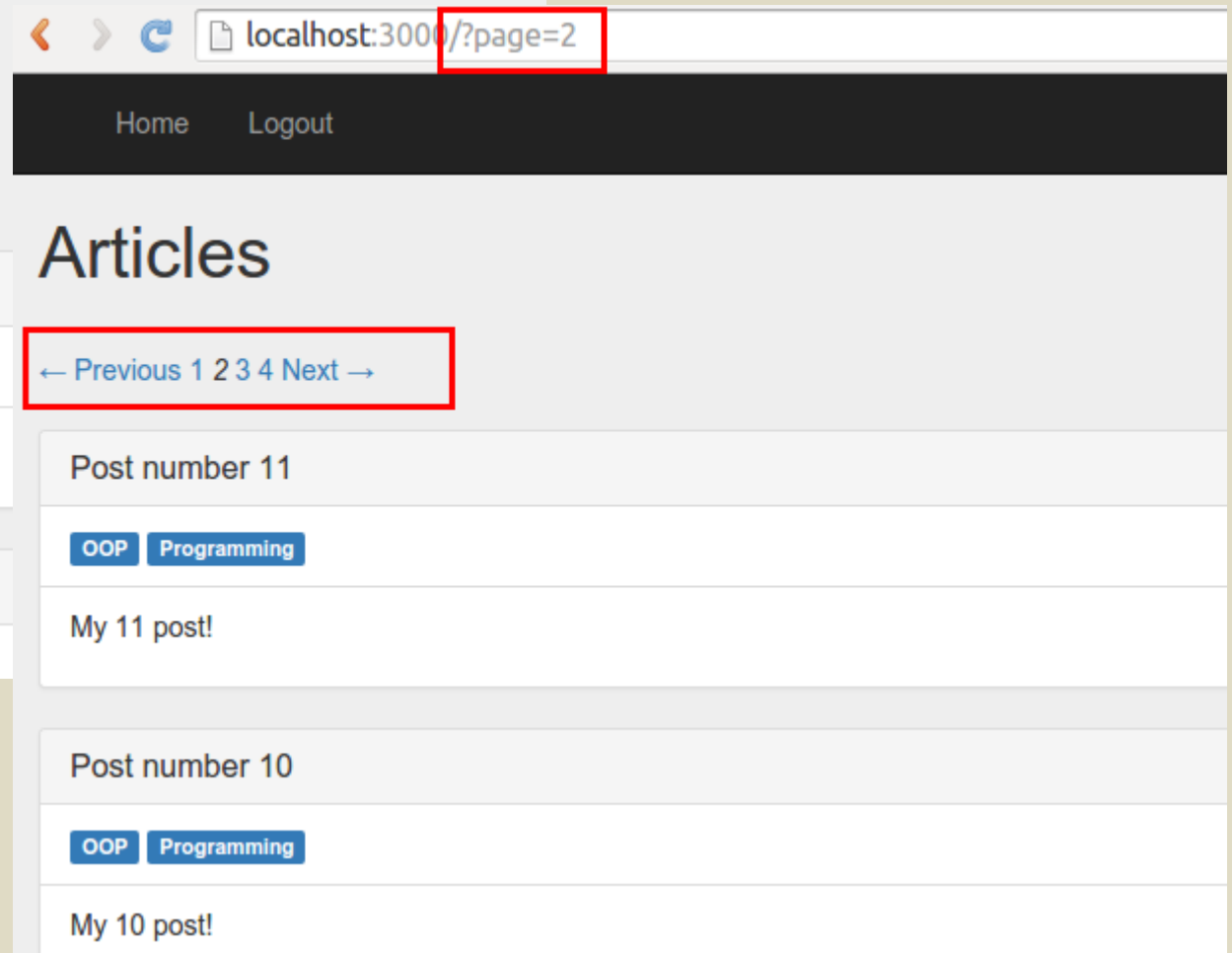
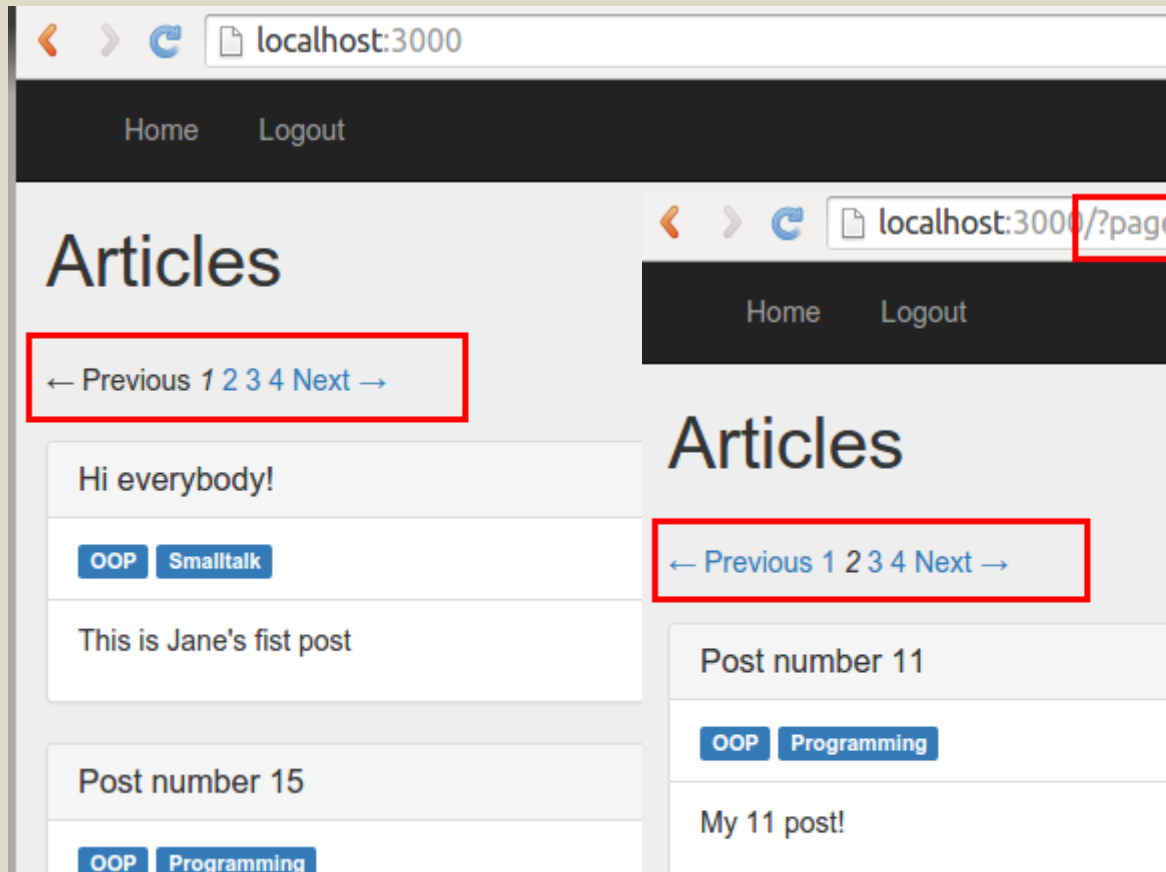
```
<h1>Articles</h1>
```

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

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

```
...
```

Will paginate



Will paginate - bootstrap

- Podemos acomodar el estilo a mano o usar una gema

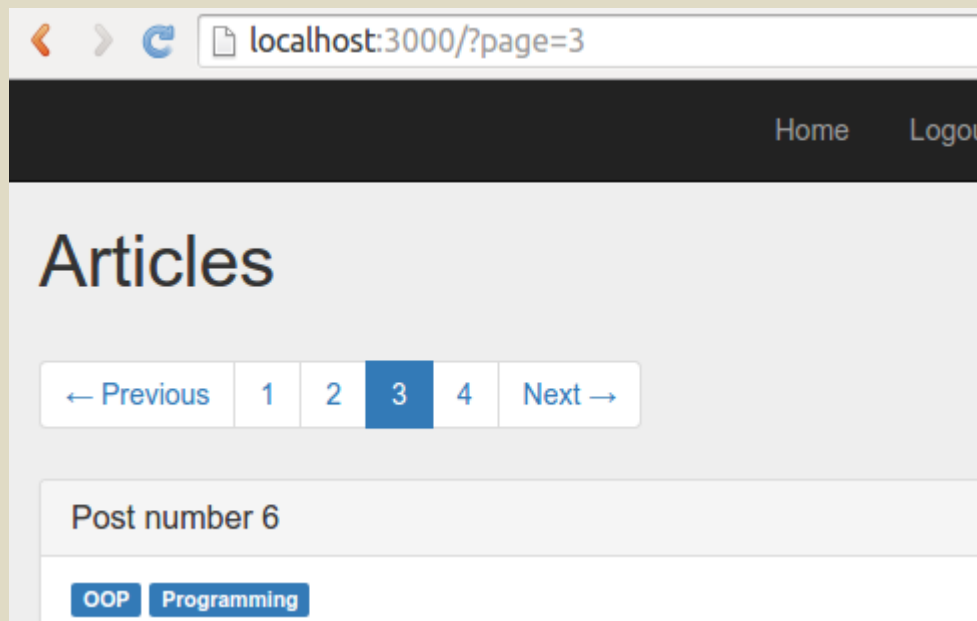
```
/Gemfile
```

```
..  
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  
      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
      where('title LIKE ?', "%#{search}%")
    else
      all
    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
      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

Orden en las búsquedas

- Links para capturar el orden.
- Modificar la búsqueda.

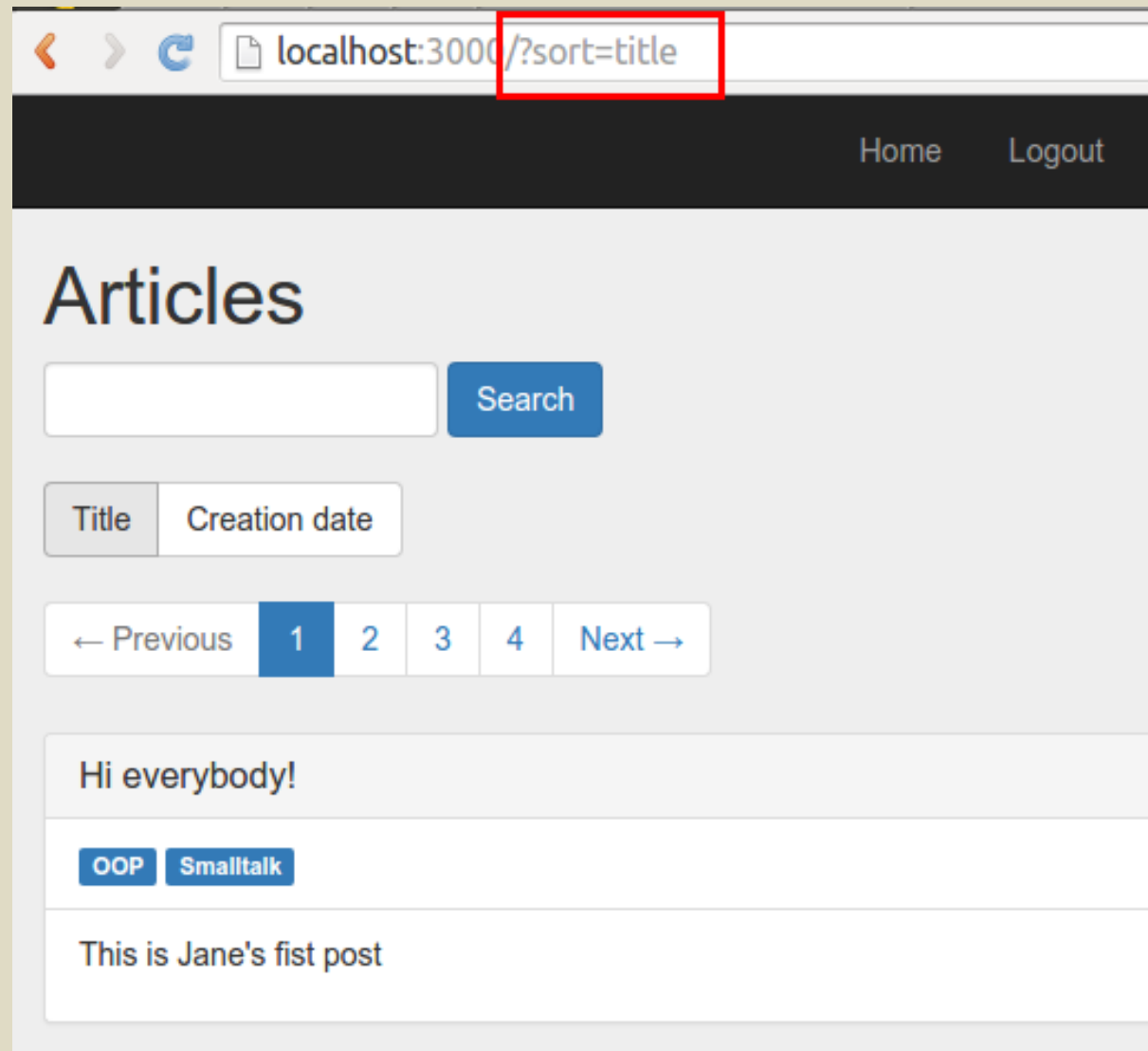


Orden en las búsquedas

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

```
...  
<div class="btn-group" role="group" style="margin-top:20px;">  
  <%= link_to "Title", {:sort => "title"}, :class => "btn btn-default" %>  
  <%= link_to "Creation date", {:sort => "created_at"}, :class => "btn btn-default" %>  
</div>  
  
<div>  
  <%= will_paginate @articles, renderer: BootstrapPagination::Rails %>  
</div>  
...
```

Orden en las búsquedas



Orden en las 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(params[:sort])
  end
  ...
end
```

Orden en las búsquedas



- Vean la salida de la consola para confirmar los queries.
- Problemas
 - Si hacemos una búsqueda se pierde el orden.
 - Si cambiamos el orden perdemos la búsqueda.
- Tenemos que propagar los parámetros existentes.

Orden en las búsquedas

- En el formulario, con un campo invisible

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

```
...  
<%= form_tag articles_path, :method => 'get', class:  
'form-inline' do %>  
  <%= hidden_field_tag :sort, params[:sort] %>  
  <div class="form-group">  
    <%= text_field_tag :search, params[:search], class:  
'form-control' %>  
    <%= submit_tag "Search", :name => nil, class: 'btn  
btn-primary' %>  
  </div>  
<% end %>  
...
```

Orden en las búsquedas

- Con parámetros en un url

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

```
...  
<div class="btn-group" role="group" style="margin-top:20px;">  
  <%= link_to "Title", params.merge(:sort => "title",  
:page => nil), :class => "btn btn-default" %>  
  <%= link_to "Creation date", params.merge(:sort =>  
"created_at", :page => nil), :class => "btn btn-default"  
%>  
</div>  
...
```


Listados

- Cosas pendientes
 - Factorización y helpers.
 - Indicar criterio de orden seleccionado.
 - Modificar criterio ascendente / descendente.

Listados

- Ver
 - <http://railscasts.com/episodes/228-sortable-table-columns>
 - <http://railscasts.com/episodes/240-search-sort-paginate-with-ajax>

Búsquedas



- Algunas gemas
 - https://github.com/wvanbergen/scoped_search
 - https://github.com/dougal/acts_as_indexed
 - <https://github.com/yzhanginwa/rails-simple-search>
 - <https://github.com/jhund/filterrific>
 - <https://github.com/activerecord-hackery/ransack>

Links



- https://github.com/mislav/will_paginate
- <https://gist.github.com/mislav/622561>
- <https://dev.mysql.com/doc/refman/5.0/en/string-comparison-functions.html>
- <http://railscasts.com/episodes/370-ransack>