



Primeros pasos en Rails

Javier Smaldone

<http://www.smaldone.com.ar>

**5tas Jornadas Regionales de
Software Libre**

Noviembre de 2005

Rosario, Santa Fe, Argentina

Problemática del desarrollo web

- Limitaciones de HTTP.
- Limitaciones de los lenguajes (Perl, PHP, etc.).
- Limitaciones de los frameworks existentes.
- Incompatibilidad del modelo relacional con el orientado a objetos.



Resultados

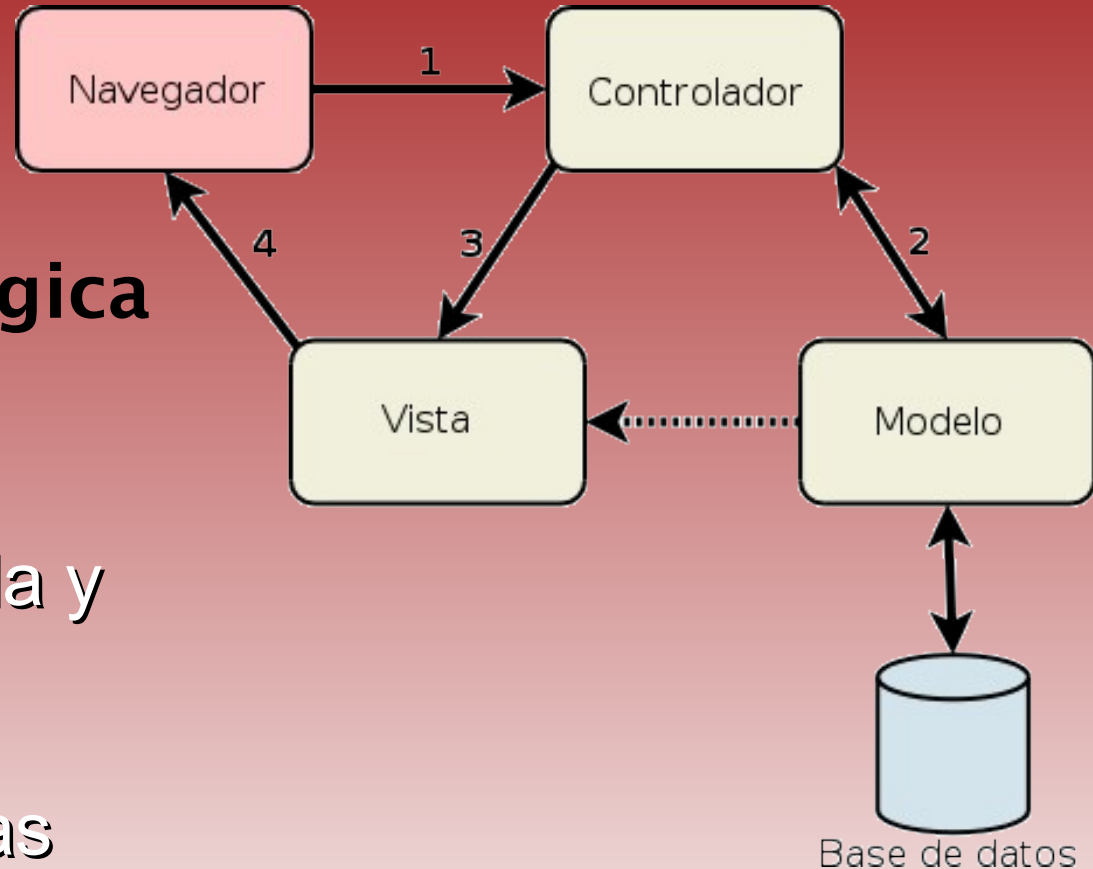
- **Duplicación de esfuerzo (y código).**
- **Aplicaciones difíciles de mantener (y escalar).**
- **Aplicaciones poco “usables”.**



El patrón MVC

- **Modelo = Estado**
- **Vista = Interfaz**
- **Controlador = Lógica**

Provee una arquitectura sólida y un nivel de abstracción apropiado para las aplicaciones web



Ruby on Rails - Origen

- David Heinemeier Hansson, un programador pragmático (“Best Hacker” 2005).
- Basecamp y Rails (el huevo y la gallina).
- Más de 15 desarrolladores en la actualidad.

¿Qué es?

- Librería de clases (componentes).
- Herramientas (scripts).



Filosofía

- Agile development
- DRY (Don't Repeat Yourself)



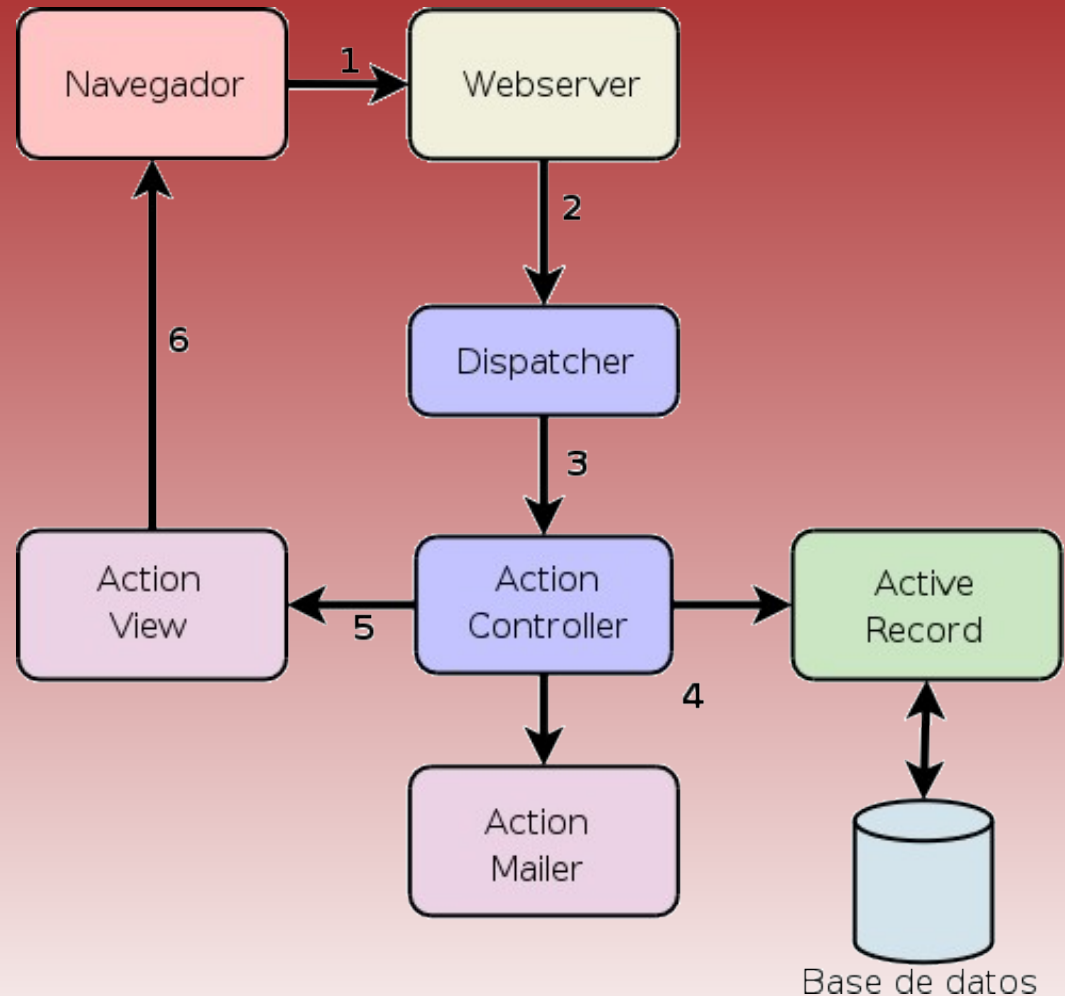
Ruby

- 100% orientado a objetos.
- Derivado de SmallTalk (altamente dinámico).
- Objetos, Clases y Metaclases.
- *“Toda diferencia entre Ruby y Perl se encuentra en Smalltalk”*
- *“Ruby es Perl pero bien hecho”*



Rails - Arquitectura

- **Action Controller**
- **Active Record**
- **Action View**
- **Action Mailer**



Active Record y las bases de datos

- **ORM (Object/Relational Mapping).**
- **Clase → Tabla**
- **SQL sólo cuando sea necesario.**



Action View (más que templates)

- **Ni Smarty ni Template-Toolkit.**
- **Ruby embebido**
- **Integración con los demás componentes.**



Ejemplo

RNews



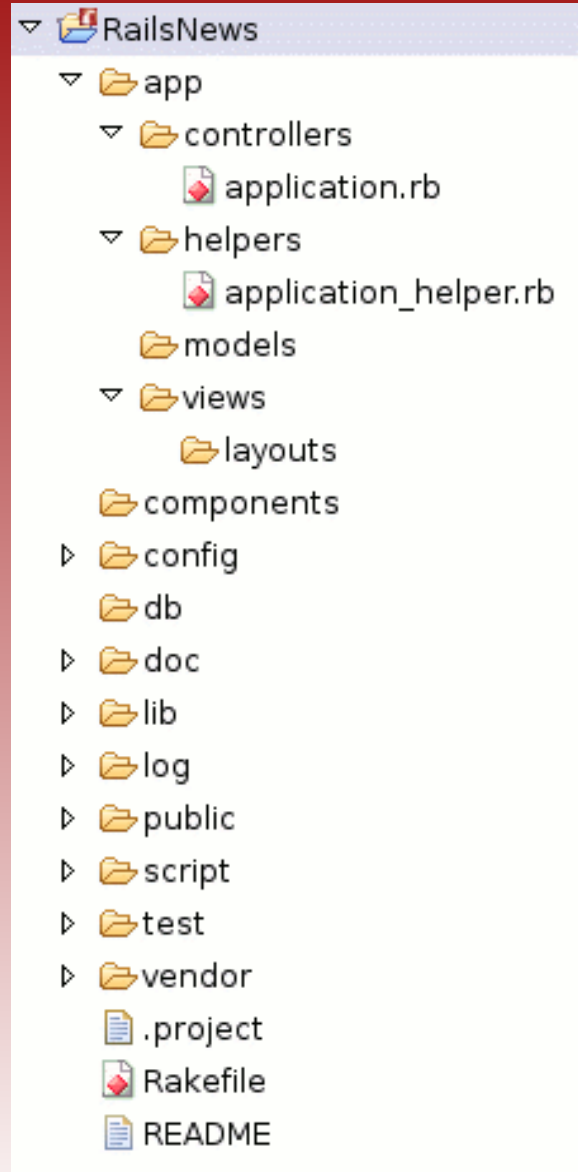
Rnews

```
javier@obelix:~/rails$ rails rnews
  create
  create  app/controllers
  create  app/helpers
  create  app/models
  create  app/views/layouts
  create  config/environments
  create  components

...
  create  public/javascripts/effects.js
  create  public/javascripts/dragdrop.js
  create  public/javascripts/controls.js
  create  doc/README_FOR_APP
  create  log/server.log
  create  log/production.log
  create  log/development.log
  create  log/test.log
javier@obelix:~/rails$ █
```



Rnews



Rnews

```
create table categories (  
  id          int          not null auto_increment,  
  title       varchar(100) not null,  
  primary key (id)  
) type=InnoDB;  
  
create table authors (  
  id          int          not null auto_increment,  
  first_name  varchar(100) not null,  
  last_name   varchar(100) not null,  
  primary key (id)  
) type=InnoDB;  
  
create table articles (  
  id          int          not null auto_increment,  
  title       varchar(100) not null,  
  content     text         not null,  
  updated_at  datetime     not null,  
  created_at  datetime     not null,  
  category_id int          not null,  
  constraint fk_a_category foreign key (category_id) references categories(id),  
  index id_articles (category_id),  
  primary key (id)  
) type=InnoDB;  
  
create table articles_authors (  
  article_id int          not null,  
  author_id  int          not null,  
  constraint fk_aa_article foreign key (article_id) references articles(id),  
  constraint fk_aa_author  foreign key (author_id)  references authors(id),  
  index id_aa_article (article_id),  
  index id_aa_author  (author_id),  
  primary key (article_id, author_id)  
) type=InnoDB;
```



Rnews

```
development:
  adapter: mysql
  database: rnews_development
  host:
  username: root
  password:

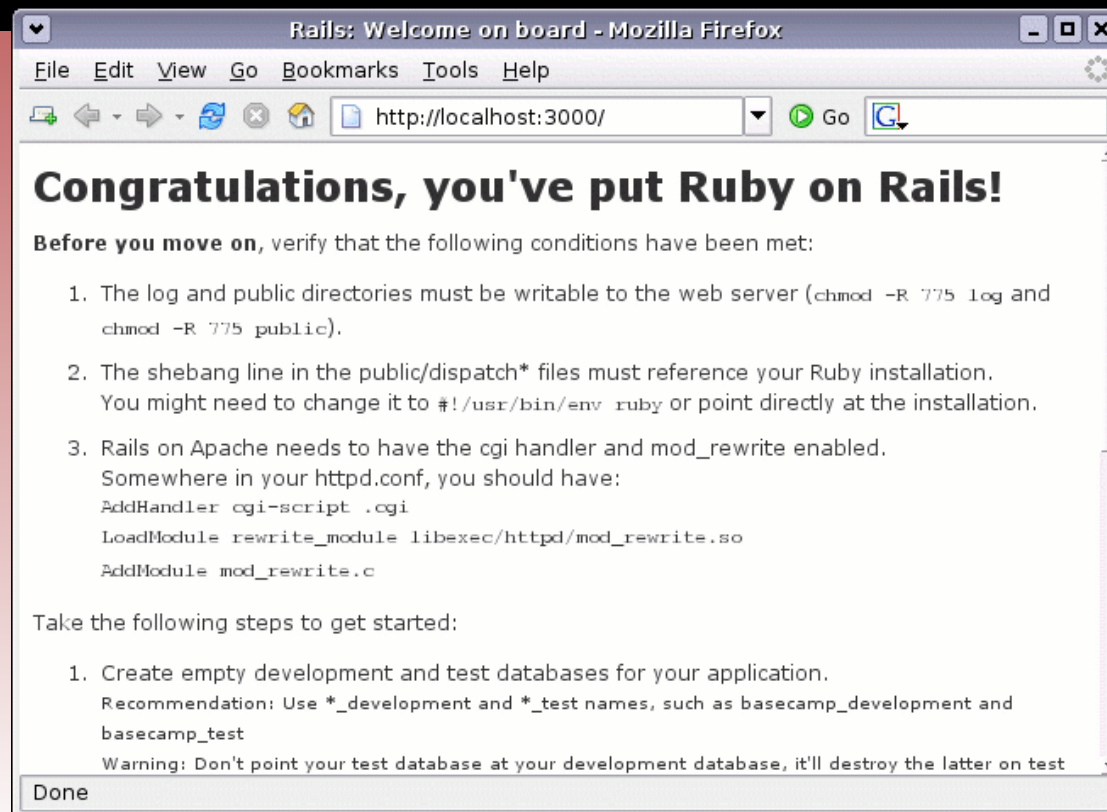
test:
  adapter: mysql
  database: rnews_test
  host:
  username: root
  password:

production:
  adapter: sqlite3
  database: ":memory:"
  host:
  username: rnews
  password:
```



Rnews

```
javier@obelix:~/rails/rnews$ script/server
=> Booting WEBrick...
=> Rails application started on http://0.0.0.0:3000
=> Ctrl-C to shutdown server; call with --help for options
[2005-11-16 17:13:11] INFO  WEBrick 1.3.1
[2005-11-16 17:13:11] INFO  ruby 1.8.4 (2005-10-29) [i486-linux]
[2005-11-16 17:13:11] INFO  WEBrick::HTTPServer#start: pid=9989 port=3000
```



Rails: Welcome on board - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost:3000/ Go

Congratulations, you've put Ruby on Rails!

Before you move on, verify that the following conditions have been met:

1. The log and public directories must be writable to the web server (`chmod -R 775 log` and `chmod -R 775 public`).
2. The shebang line in the `public/dispatch*` files must reference your Ruby installation. You might need to change it to `#!/usr/bin/env ruby` or point directly at the installation.
3. Rails on Apache needs to have the `cgi` handler and `mod_rewrite` enabled. Somewhere in your `httpd.conf`, you should have:

```
AddHandler cgi-script .cgi
LoadModule rewrite_module libexec/httpd/mod_rewrite.so
AddModule mod_rewrite.c
```

Take the following steps to get started:

1. Create empty development and test databases for your application.
Recommendation: Use `*_development` and `*_test` names, such as `basecamp_development` and `basecamp_test`
Warning: Don't point your test database at your development database, it'll destroy the latter on test

Done



Rnews

```
javier@obelix:~/rails/rnews$ ruby script/generate model Category
exists  app/models/
exists  test/unit/
exists  test/fixtures/
create  app/models/category.rb
create  test/unit/category_test.rb
create  test/fixtures/categories.yml
javier@obelix:~/rails/rnews$ ruby script/generate model Article
exists  app/models/
exists  test/unit/
exists  test/fixtures/
create  app/models/article.rb
create  test/unit/article_test.rb
create  test/fixtures/articles.yml
javier@obelix:~/rails/rnews$ ruby script/generate model Author
exists  app/models/
exists  test/unit/
exists  test/fixtures/
create  app/models/author.rb
create  test/unit/author_test.rb
create  test/fixtures/authors.yml
javier@obelix:~/rails/rnews$
```



Rnews

```
class Category < ActiveRecord::Base  
  has_many :articles  
  validates_presence_of :title  
  
end
```

```
class Author < ActiveRecord::Base  
  has_and_belongs_to_many :articles  
  validates_presence_of :first_name, :last_name  
  
end
```

```
class Article < ActiveRecord::Base  
  belongs_to :category  
  has_and_belongs_to_many :authors  
  validates_presence_of :title, :content  
  
end
```



Rnews

```
javier@obelix:~/rails/rnews$ ruby script/generate scaffold Category
  exists  app/controllers/
  exists  app/helpers/
  create  app/views/categories
  exists  test/functional/
dependency model
  exists  app/models/
  exists  test/unit/
  exists  test/fixtures/
  skip    app/models/category.rb
  identical test/unit/category_test.rb
  identical test/fixtures/categories.yml
  create  app/views/categories/_form.rhtml
  create  app/views/categories/list.rhtml
  create  app/views/categories/show.rhtml
  create  app/views/categories/new.rhtml
  create  app/views/categories/edit.rhtml
  create  app/controllers/categories_controller.rb
  create  test/functional/categories_controller_test.rb
  create  app/helpers/categories_helper.rb
  create  app/views/layouts/categories.rhtml
  create  public/stylesheets/scaffold.css
javier@obelix:~/rails/rnews$ ruby script/generate scaffold Author
javier@obelix:~/rails/rnews$ ruby script/generate scaffold Article
```



Rnews

```
class CategoriesController < ApplicationController
  def index
    list
    render :action => 'list'
  end

  def list
    @category_pages, @categories = paginate :categories, :per_page => 10
  end

  def show
    @category = Category.find(params[:id])
  end

  def new
    @category = Category.new
  end

  def create
    @category = Category.new(params[:category])
    if @category.save
      flash[:notice] = 'Category was successfully created.'
      redirect_to :action => 'list'
    else
      render :action => 'new'
    end
  end
end
```



Rnews

```
def edit
  @category = Category.find(params[:id])
end

def update
  @category = Category.find(params[:id])
  if @category.update_attributes(params[:category])
    flash[:notice] = 'Category was successfully updated.'
    redirect_to :action => 'show', :id => @category
  else
    render :action => 'edit'
  end
end

def destroy
  Category.find(params[:id]).destroy
  redirect_to :action => 'list'
end
end
```



Rnews

Authors: index - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost:3000/authors

Listing authors

First name Last name

[New author](#)

Done

Authors: new - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost:3000/authors/new

New author

First name
Edsger

Last name
Dijkstra

[Back](#)

Done



Rnews

Authors: edit - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost:3000/authors/edit/1

Editing author

First name

Last name

[Show](#) | [Back](#)

Done

RNews - authors::create - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost:3000/authors/create

New author

1 error prohibited this author from being saved

There were problems with the following fields:

- Last name can't be blank

First name

Last name

[Back](#)

Done



Rnews

```
<%= error_messages_for 'article' %>

<p><label for="article_title">Title</label><br/>
<%= text_field 'article', 'title' %></p>

<p><label for="article_content">Content</label><br/>
<%= text_area 'article', 'content' %></p>

<!-- <p><label for="article_updated_at">Updated at</label><br/>
<%= datetime_select 'article', 'updated_at' %></p>

<p><label for="article_created_at">Created at</label><br/>
<%= datetime_select 'article', 'created_at' %></p> -->

<p><label for="article_category_id">Category</label><br />
<select id="article_category_id" name="article[category_id]">
  <%= for category in Category.find(:all) %>
    <option value="<%= category.id %>" <%= 'selected' if category.id == @article.category_id %>>
      <%= category.title %>
    </option>
  <%= end %>
</select>
</p>

<p><label for="article[author_ids][]">Authors</label><br/>
<%= for author in Author.find(:all) %>
  <%= check_box_tag("article[author_ids][]" ,
                    author.id,
                    @article.authors.include?(author)) %>
  <%=h author.first_name+' '+author.last_name %><br/>
<%= end %></p>
```



Rnews

```
def update
  @article = Article.find(params[:id])

  if !params['article']['author_ids']
    @article.authors.clear
  end

  if @article.update_attributes(params[:article])
    flash[:notice] = 'Article was successfully updated.'
    redirect_to :action => 'show', :id => @article
  else
    render :action => 'edit'
  end
end
```



Rnews

Articles: new - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost:3000/articles/new

New article

Title

Content

```
When the idea to write about the next
fifty years of computing first entered
my head, I wrote it off as utterly
preposterous: what sane scientist
purports to be able to see so far into
the future? But then I realized that in
a way that is precisely what educators
do all the time: when designing our
courses, we do dare to decide what to
teach and what to ignore, and we do this
for the benefit of students, many of
whom will still be active forty to fifty
years from now. Clearly some vision of
the next half century of computing
science is operational. To this I should
add that it is all right if the crystal
ball is too foggy to show much detail.
Thirty-five years ago, for instance, I
had no inkling of how closely program
design and proof design would come
together, and in such a detailed sense
```

Category

Authors

Edsger Dijkstra

Donald Knuth

[Back](#)

Done



Rnews

```
javier@obelix:~/rails/rnews$ ruby script/generate controller Admin
exists  app/controllers/
exists  app/helpers/
create  app/views/admin
exists  test/functional/
create  app/controllers/admin_controller.rb
create  test/functional/admin_controller_test.rb
create  app/helpers/admin_helper.rb
javier@obelix:~/rails/rnews$
```

```
class AdminController < ApplicationController
  def index
  end
end
```



Rnews

```
<h1>Administration</h1>
```

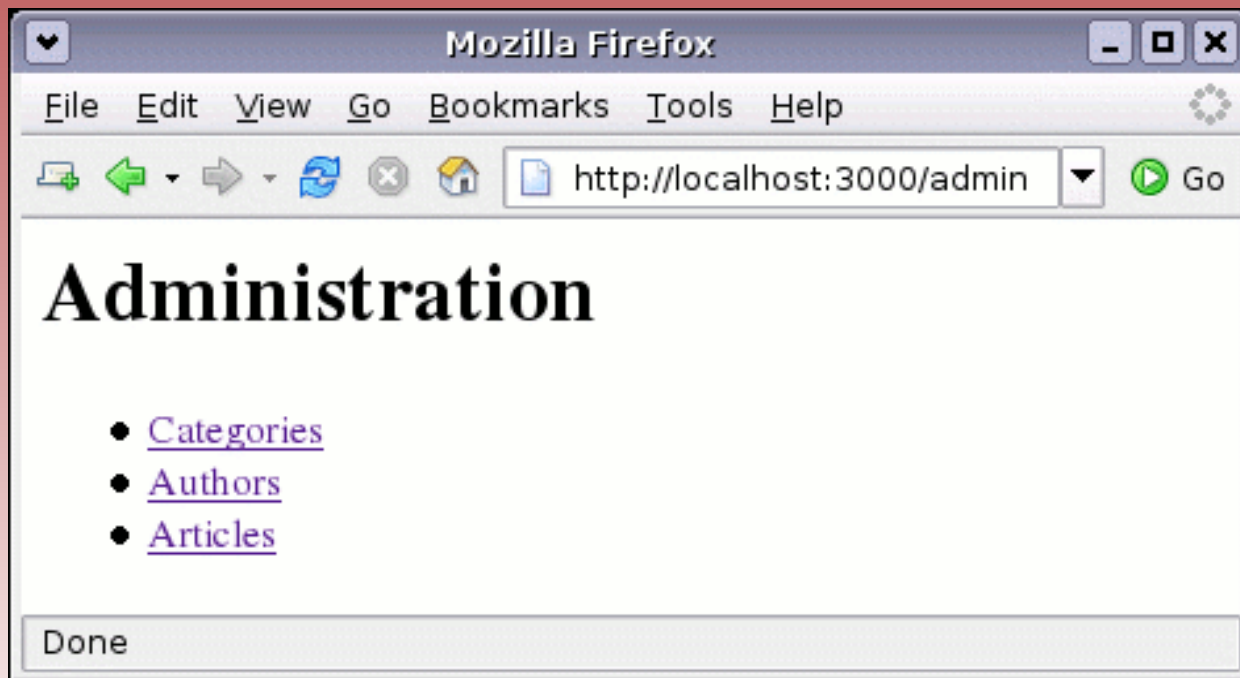
```
<ul>
```

```
<li><%= link_to 'Categories', :controller => 'categories', :action => 'list' %>
```

```
<li><%= link_to 'Authors', :controller => 'authors', :action => 'list' %>
```

```
<li><%= link_to 'Articles', :controller => 'articles', :action => 'list' %>
```

```
</ul>
```

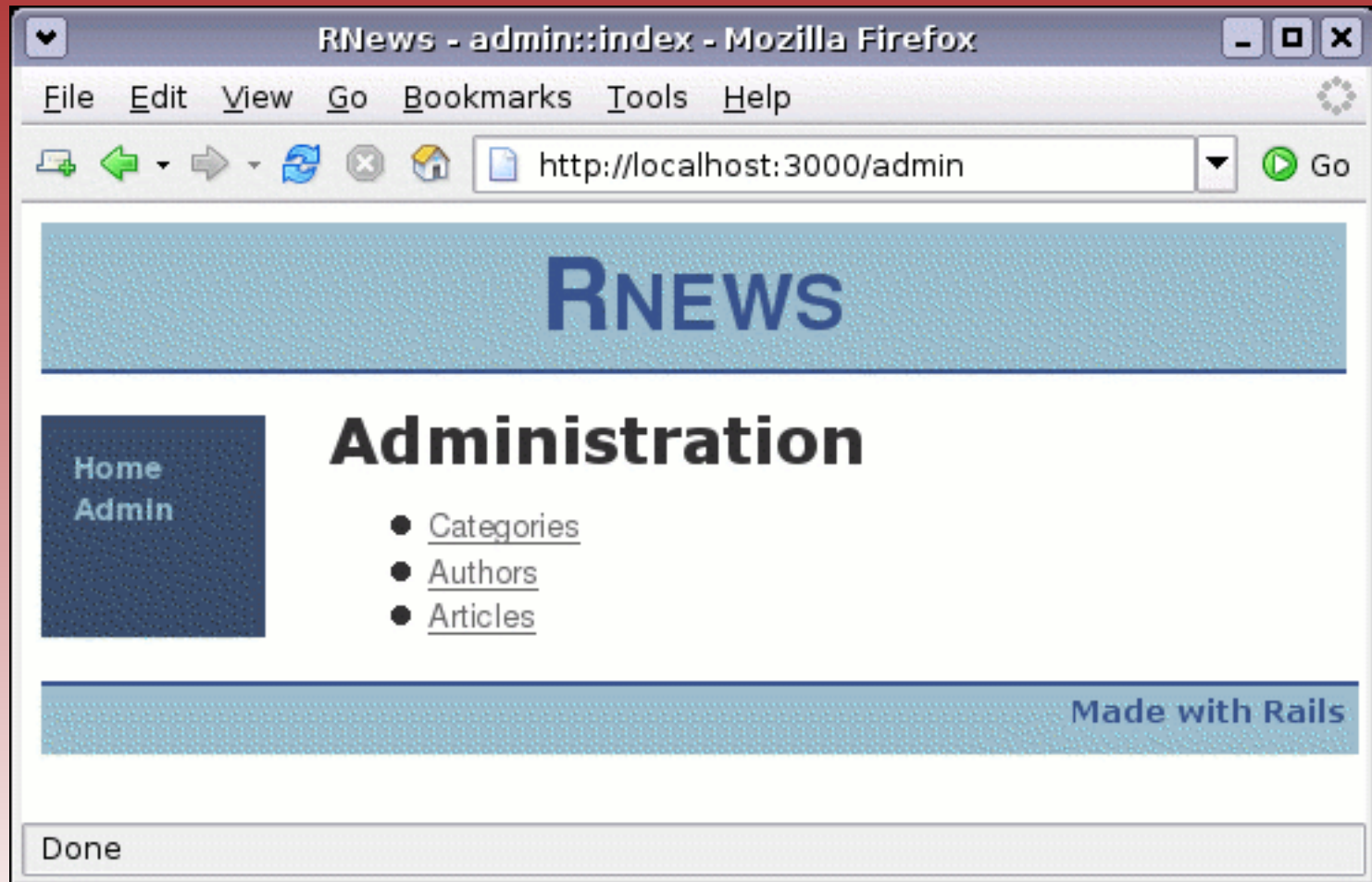


Rnews

```
<html>
  <head>
    <title>RNews - <%= controller.controller_name %>::
    <%= controller.action_name %></title>
    <%= stylesheet_link_tag "scaffold", "screen", :media => "all" %>
  </head>
  <body>
    <div id="banner">Rnews</div>
    <div id="columns">
      <div id="side">
        <%= link_to 'Home', :controller => 'home' %><br />
        <%= link_to 'Admin', :controller => 'admin' %><br />
      </div>
      <div id="main">
        <%= if @flash[:notice] -%>
          <div id="notice"><%= @flash[:notice] %></div>
        <%= end -%>
        <%= @content_for_layout %>
      </div>
    </div>
    <div id="endbanner">Made with Rails</div>
  </body>
</html>
```



Rnews



Rnews

RNews - home::index - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://localhost:3000/home

RNEWS

Home
Admin

Articles

Computer Science

The next fifty years

When the idea to write about the next fifty years of computing first entered my head, I wrote it off as utterly preposterous: what sane scientist purports to be able to see so far into the future? B...

[Show](#)

Made with Rails

Done



Rnews

```
javier@obelix:~/rails/rnews$ rake stats  
(in /home/javier/rails/rnews)
```

Name	Lines	LOC	Classes	Methods	M/C	LOC/M
Helpers	13	12	0	0	0	0
Controllers	162	135	6	26	4	3
Components	0	0	0	0	0	0
Functional tests	300	215	10	36	3	3
Models	22	13	3	0	0	0
Unit tests	30	21	3	3	1	5
Libraries	0	0	0	0	0	0
Total	527	396	22	65	2	4
Code LOC: 160 Test LOC: 236 Code to Test Ratio: 1:1.5						



Rnews

Rails Application Documentation - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

file:///home/javier/rails/rnews/doc/app/index.html

Files	Classes	Methods
app/controllers/a	AdminController	create (CategoriesController)
app/controllers/a	AdminHelper	create (ArticlesController)
app/controllers/a	ApplicationController	create (AuthorsController)

Public Instance methods

create()

[\[Source\]](#)

```
# File app/controllers/categories_controller.rb, line 19
19: def create
20:   @category = Category.new(params[:category])
21:   if @category.save
22:     flash[:notice] = 'Category was successfully created.'
23:     redirect_to :action => 'list'
24:   else
25:     render :action => 'new'
26:   end
27: end
```

file:///home/javier/rails/rnews/doc/app/classes/CategoriesController.html#



Potencialidad de RoR

- **Bases de datos complejas:**
 - **Bloqueo optimista.**
 - **Transacciones.**
 - **Sentencias SQL.**
- **Testing**
- **AJAX (Web 2.0)**



Enlaces y material de referencia

- Sitio de RoR: <http://www.rubyonrails.org/>
- *Agile Web Development with Rails. A pragmatic Guide.* Dave Thomas y David Heinemeier Hansson.
<http://www.pragmaticprogrammer.com/titles/rails/>
- Sitio del lenguaje Ruby:
<http://www.ruby-lang.org/>



Preguntas y respuestas





¡Muchas gracias!

Javier Smaldone

<http://www.smaldone.com.ar>

**5tas Jornadas Regionales de
Software Libre**

Noviembre de 2005

Rosario, Santa Fe, Argentina