



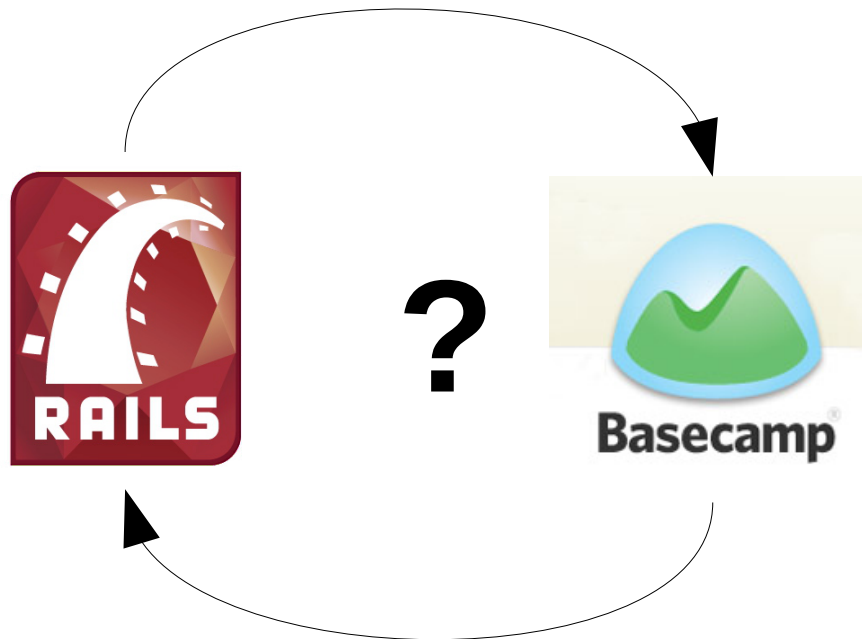
FLISOL 2015 -CABA-

Introducción a Ruby on Rails

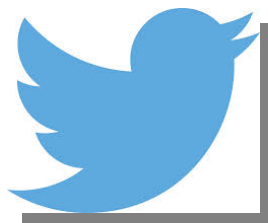
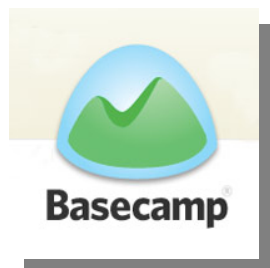
Diego Gomez – diego@dgomez.com.ar



En el principio...



Da vida a



Framework



- Marco de trabajo
- Estructura conceptual
- Arquitectura



Rails



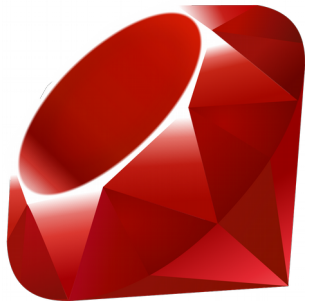
Framework escrito en
Ruby para desarrollar
aplicaciones web en *Ruby*.



Ruby



- Multiplataforma
- Dinámico / Interpretado
- Buenas influencias (SmallTalk, Lisp, Eiffel, Java)
- Orientado a objetos (100%)
- Introspección, reflexión y metaprogramación
- Expresiones Regulares (simil Perl)
- Extensiones, librerías, soporte



Ruby (Classes)



```
class MyPerson

  current_time = Time.now

  def name
    @name
  end

  def name=(name)
    @name = name
  end

  def is_married
    ...
  end
end
```

```
Irb>p = MyPerson.new
Irb>p.name = 'Alice'
Irb>puts p.name
=> "Alice"
```



Ruby (Métodos)



```
class MyPerson
  def m1 #this method is public
    ...
  end

  protected
  def m2 #this methos is protected
    ...
  end

  private
  def m3 #this methos is private
    ...
  end
end
```



Ruby (arrays)



```
irb>a = [1, 'cat', 2, 3.1416]
```

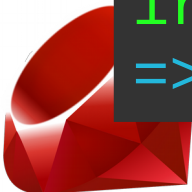
```
irb>a[1]  
=> "cat"
```

```
irb>a.reverse  
=> [3.1416, 2, "cat", 1]
```

```
irb>a.length  
=> 4
```

```
irb>a[5]  
=> nil
```

```
irb>a[2] = nil  
=> [1, "cat", nil, 3.1416]
```



Ruby (hashes)



```
tipo_instrumento = {  
  cello: 'cuerdas',  
  clarinete: 'viento',  
  piano: 'cuerdas'  
}
```

```
irb>tipo_instrumento[:cello]  
=> "cuerdas"  
irb>tipo_instrumento[:clarinete] = 'bronce'  
{:cello=>"cuerdas", :clarinete=>"bronce",  
:piano=>"cuerdas"}
```

NOTA

```
redirect_to action: 'edit', id: user.id
```



Ruby (control)



```
if count > 10
  puts "Try again"
elsif tries == 3
  puts "You lose"
else
  puts "Enter a number"
end
```

```
while weight < 100 and num_pallets <= 30
  pallet = next_pallet()
  weight += pallet.weight
  num_pallets += 1
end
```



Ruby (control 2)



```
puts "Alert!" if radiation > 3000
```

```
distance = distance * 1.2 while distance < 100
```

```
3.times {print "jo! "}  
=> jo! jo! jo!
```





Ruby (Closures)

- Funciones lambda, agentes
- Permiten paso de funciones o *bloques* como parámetro
- Naturales y transparentes
- Code Blocks: Código delimitados por `{ }` o por *do end*.

```
{ puts "hola mundo!" } # Code block
```

```
do  
  user.name = "Alice"  
  user.save  
end
```

```
saludo("Bob", "estimado usuario") {puts "Hola"}
```





Ruby (Closures)

Imprimir elementos de un arreglo

```
animales = %w(perro gato hormiga paloma)
animales.each {|animal| puts animal}
```

Multiplicar por 2 los elementos de una lista

```
def double(a)
  a.map {|e| e*2}
end
```

Calcular la edad promedio de las personas mayores de edad

```
adults = people.select{|p| p.age > 18}
sum = 0
adults.each {|p| sum += p.age }
puts sum.to_f/adults.size if adults.size > 0
```



Rails



- Ruby
- Arquitectura MVC
- DRY (Don't repeat yourself)
- Convention over Configuration
- Extensiones (gemas)
- Ambiente de trabajo completo



Rails



- Versión actual 4.2
- Ruby > 1.9 (2.2 recomendado)

```
#gem install rails
```



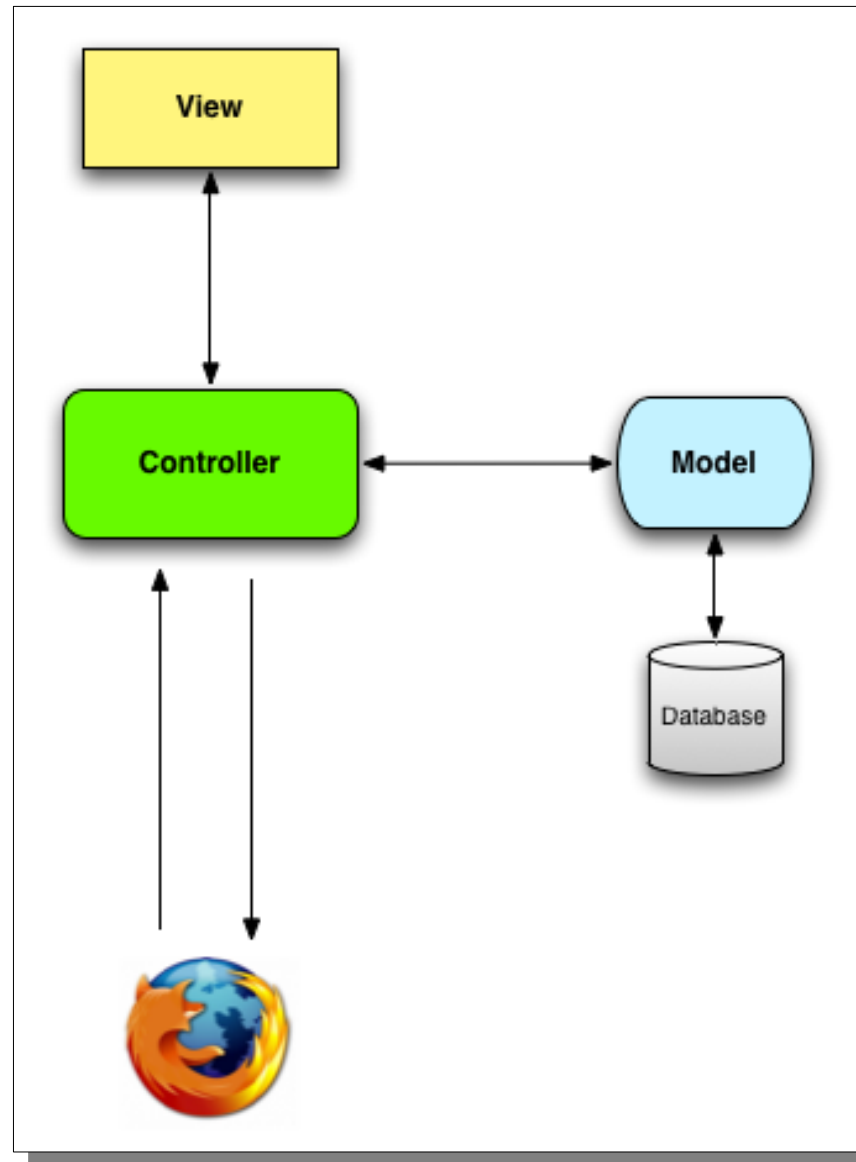
Rails (Componentes)



- Active Record (M)
- Active View (V)
- Active Controller (C)



Rails (MVC)

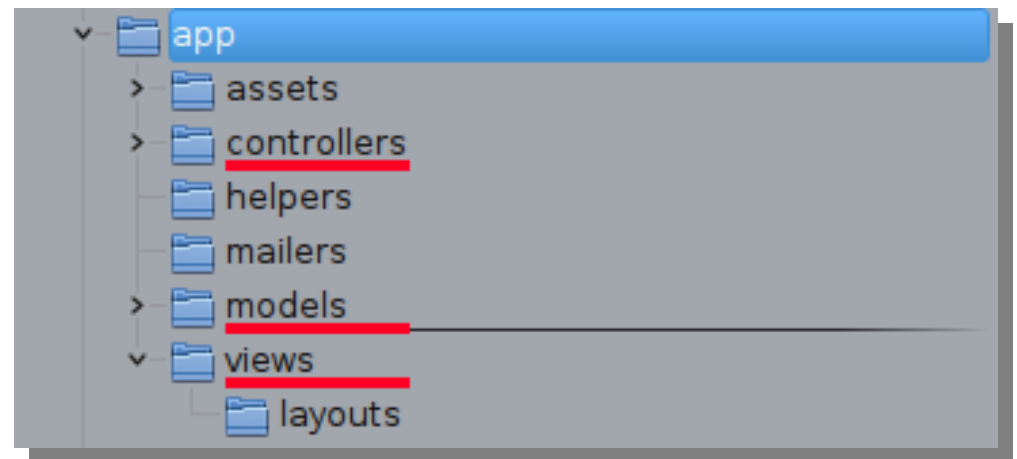
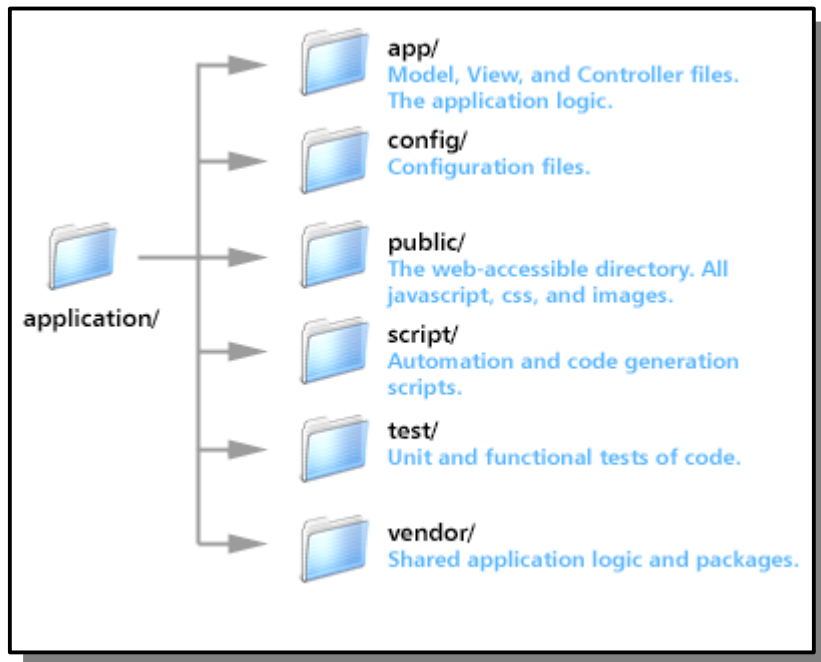


Rails



Creación proyecto

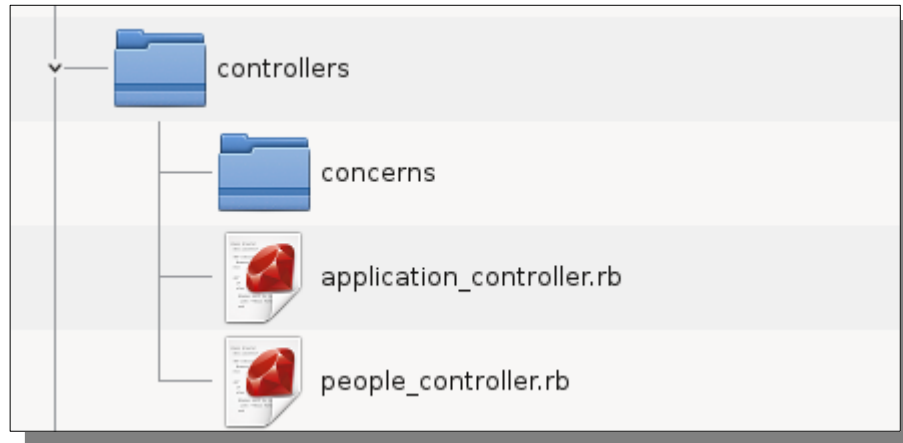
```
$rails new <nombre_app>
```



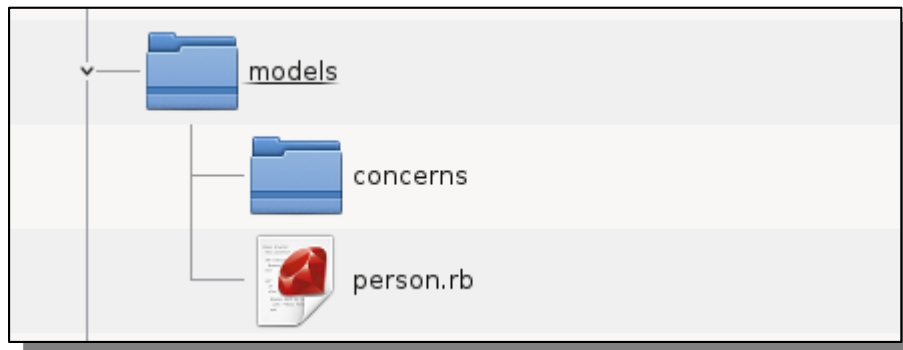
Rails



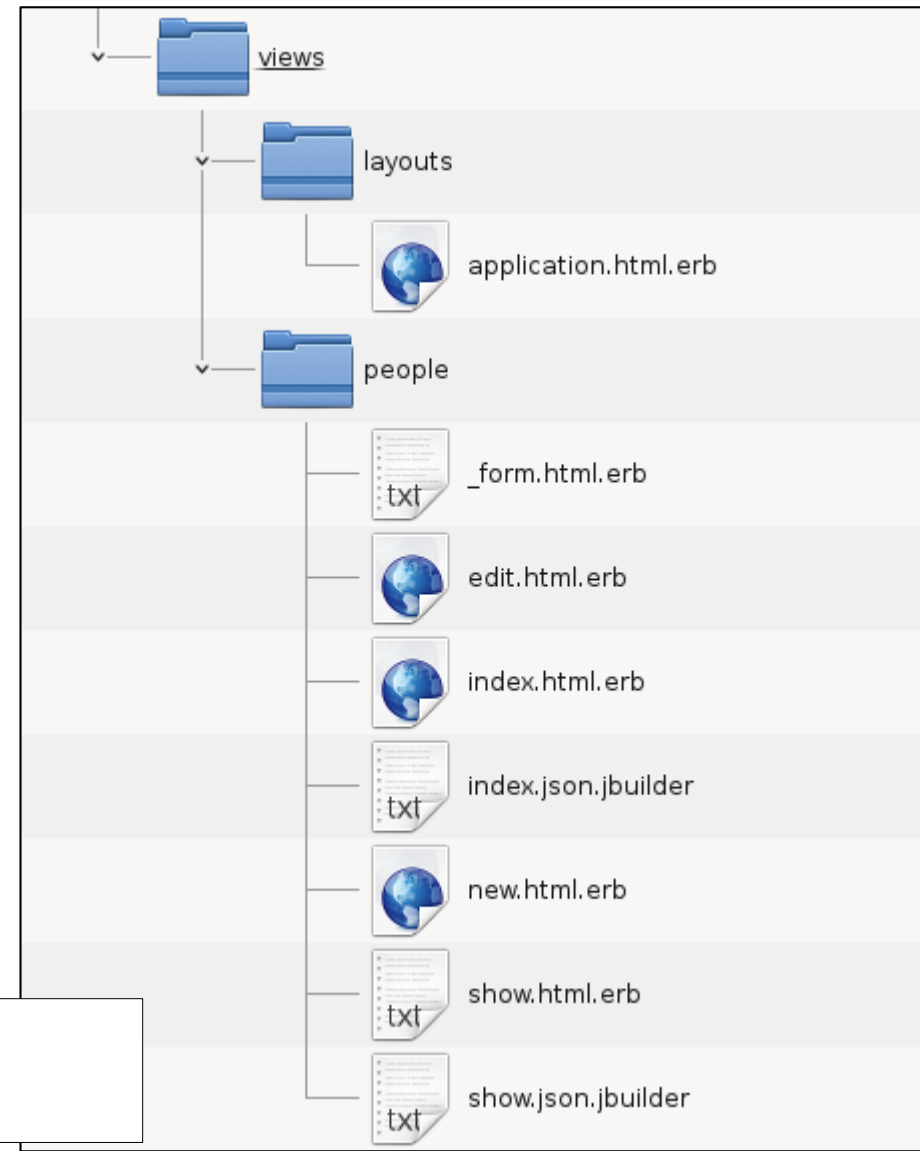
Controladores



Modelos



Vistas



<http://dom.net/people/index>
<http://dom.net/people/show/1>

Rails (Modelos)



- Active Record
- ORM
- Responsabilidad sobre los datos
- CoC
- Nombre de clase (en singular)
- Filas => Instancias
- Columnas => atributos

Tabla People

id	name	lastname	age
1	Johon	Benn	25
2	Bob	Schiff	15
3	Alice	Tylor	32

Modelo (app/model/person.rb)

```
class Person < ActiveRecord::Base  
end
```



Rails (Modelos)



Tabla People

id	name	lastname	age
1	John	Benn	25
2	Bob	Schiff	15
3	Alice	Tylor	32

Modelo (app/model/person.rb)

```
class Person < ActiveRecord::Base  
end
```

CRUD

```
p = Person.new  
p.name = 'Pepe'  
p.lastname = 'Pepe'  
p.age = '26'  
p.save
```

```
p = Person.find(1)  
puts p.name #John
```

```
p = Person.find(2)  
p.age = 45  
p.save
```

```
p = Person.find(3)  
p.destroy
```



Rails (Modelos)



Tabla People

id	name	lastname	age
1	John	Benn	25
2	Bob	Schiff	15
3	Alice	Tylor	32

Modelo (app/model/person.rb)

```
class Person < ActiveRecord::Base
  validates :name, :lastname, :age, presence: true
  validates :age, numericality: {greater_or_equal_to: 1}
  validates :id, uniqueness: true
end
```



Rails (Modelos)



Depot

localhost:3000/products/new

New product

Title

Description

<p> CoffeeScript is JavaScript done right. It provides all of JavaScript's functionality wrapped in a cleaner, more succinct syntax. In the first book on this exciting new language, CoffeeScript guru Trevor Burnham shows you how to

Image url

Price

Create Product

[Back](#)

Depot

localhost:3000/products

New product

3 errors prohibited this product from being saved:

- Title can't be blank
- Description can't be blank
- Image url can't be blank

Title

Description

Image url

Price

Create Product

[Back](#)



Rails (Controladores)



app/controllers/people_controller.rb

```
class PeopleController < ApplicationController
  def index
    @people = Person.all
    respond_to do |format|
      format.html #index.html.erb
      format.xml {render :xml => @people}
    end
  end

  def show
    @person = Person.find(params[:id])
    respond_to do |format|
      format.html #show.html.erb
      format.xml {render :xml => @person}
    end
  end
end
```

<http://localhost/people/index>
<http://localhost/people/show/1>

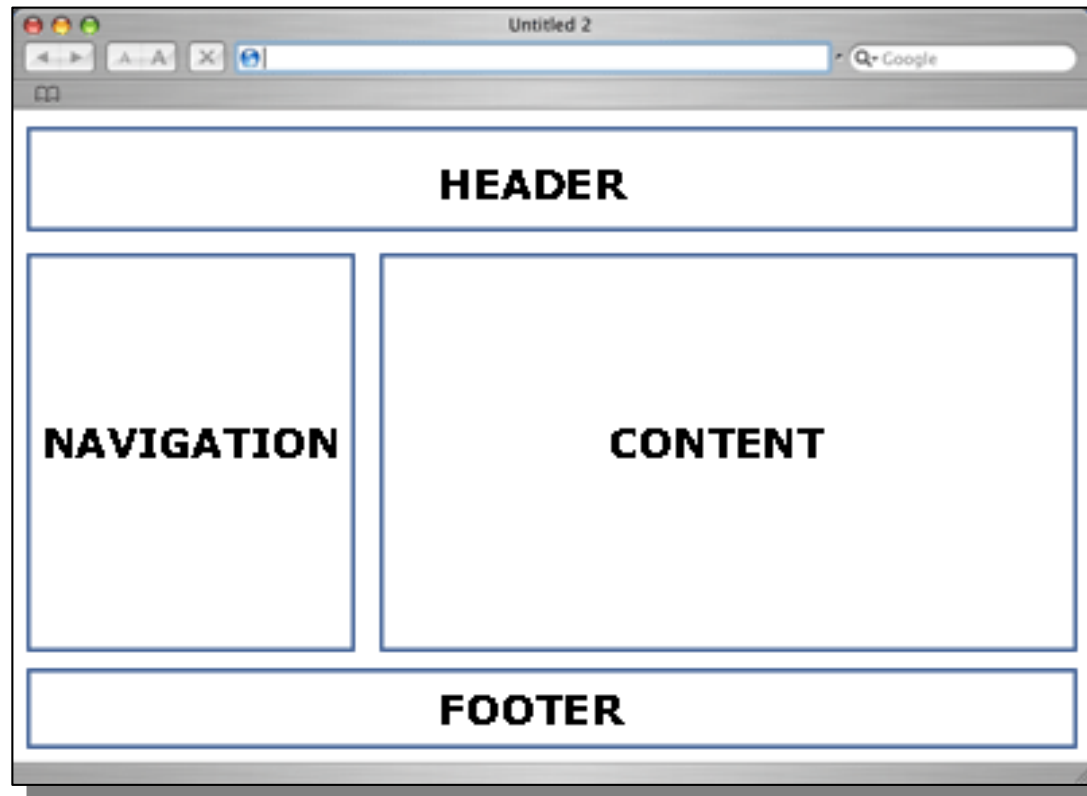
Rails (Vistas -Layout-)



app/views/layouts/application.html.erb

```
<!DOCTYPE html>
<html> <head> <title>Users</title>
  <%= stylesheet_link_tag "scaffold" %>
  <%= csrf_meta_tag %>
</head>
<body id="store">
  <%= image_tag("logo.png") %>
  </div>
  <div id="side">
    <% if session[:user_id] %>
      <%= link_to 'Users', users_path %><br />
      <%= button_to 'Logout', logout_path, :method => :delete%>
    <% end %>
  </div>
  <div id="main">
    <%= yield %>
  </div>
</div>
</body>
</html>
```

Rails (Vistas -Layout-)



Rails (Vistas)



app/views/people/index.html.erb)

```
<table>
  <% @people.each do |person| %>
    <tr>
      <td>
        <dl>
          <dt><%= person.name %></dt>
          <dd><%= truncate(strip_tags(person.lastname),
                        :length => 80) %></dd>
        </dl>
      </td>
      <td class="list_actions">
        <%= link_to 'Show', person %><br/>
        <%= link_to 'Destroy', person,
                  :confirm => 'Are you sure?',
                  :method => :delete %>
      </td>
    </tr>
  <% end %>
</table>
```

Rails (Vistas)



app/views/people/show.html.erb)

```
<p>
  <b> Name:</b>
  <%= @person.name %>
</p>

<p>
  <b>Lastname:</b>
  <%= @person.lastname %>
</p>

<p>
  <b>Age:</b>
  <%= @person.age %>
</p>

<%= link_to 'Edit', edit_person_path(@person) %> |
<%= link_to 'Back', people_path %>
```

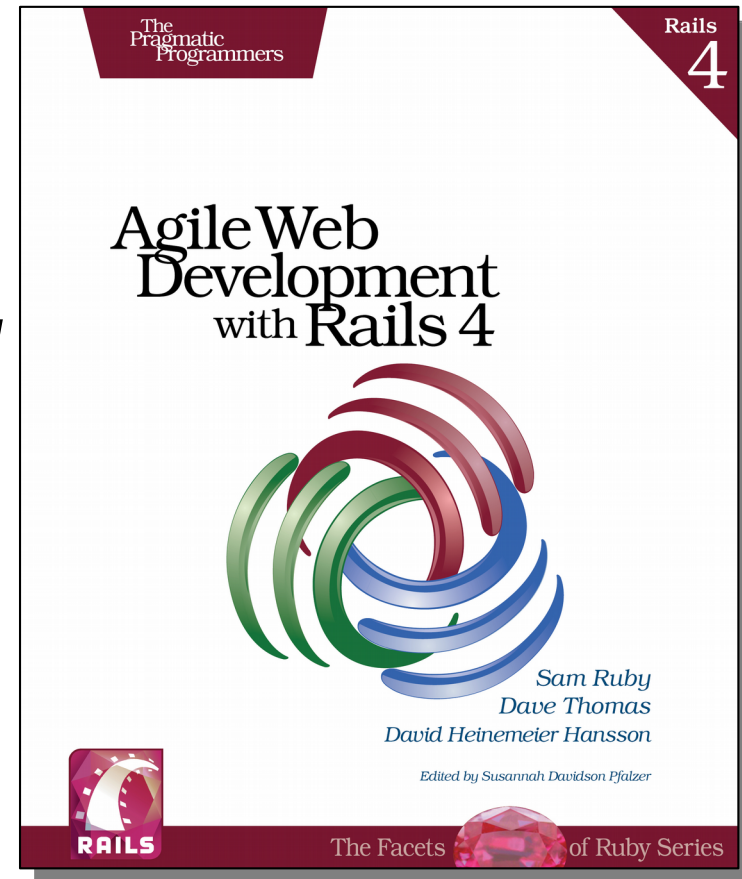


Ruby on Rails



¿Por dónde seguir?

- <http://rubyonrails.org/>
- <http://rubyonrails.org/documentation/>
- <http://api.rubyonrails.org/>



Ruby on Rails



¿Preguntas?



Ruby on Rails



¡Gracias!

