

Cómo programar un blog **REST**

Javier Vidal Postigo

<http://javiervidal.net/>

Conferencia Rails 2007



¿Qué es Rails?

**framework `ruby` para desarrollar aplicaciones
web con acceso a base de datos**

framework **ruby** para desarrollar aplicaciones
web con acceso a base de datos

convención mejor que configuración

¿Qué es MVC?

MVC en Rails



Enrutador

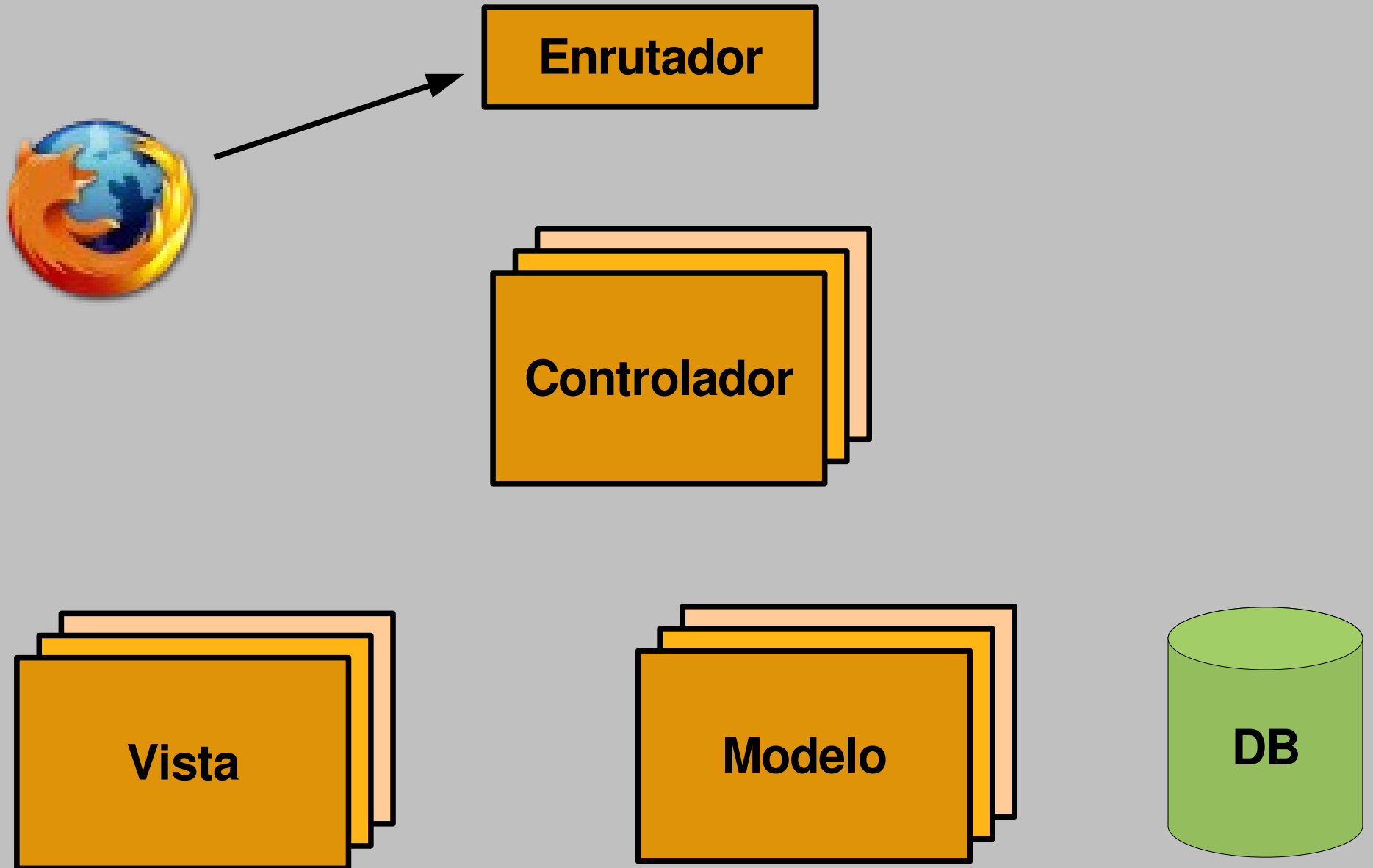
Controlador

Vista

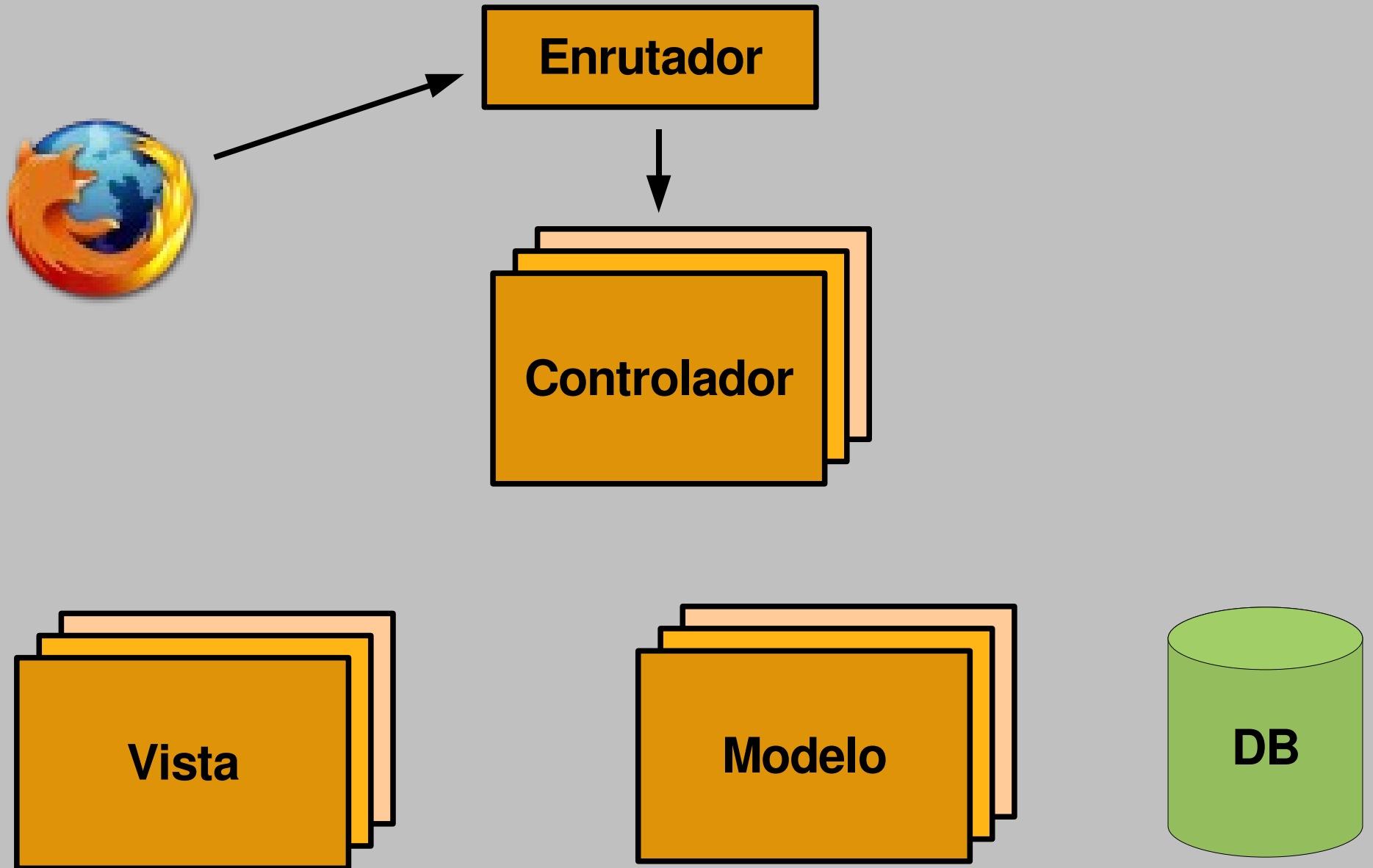
Modelo

DB

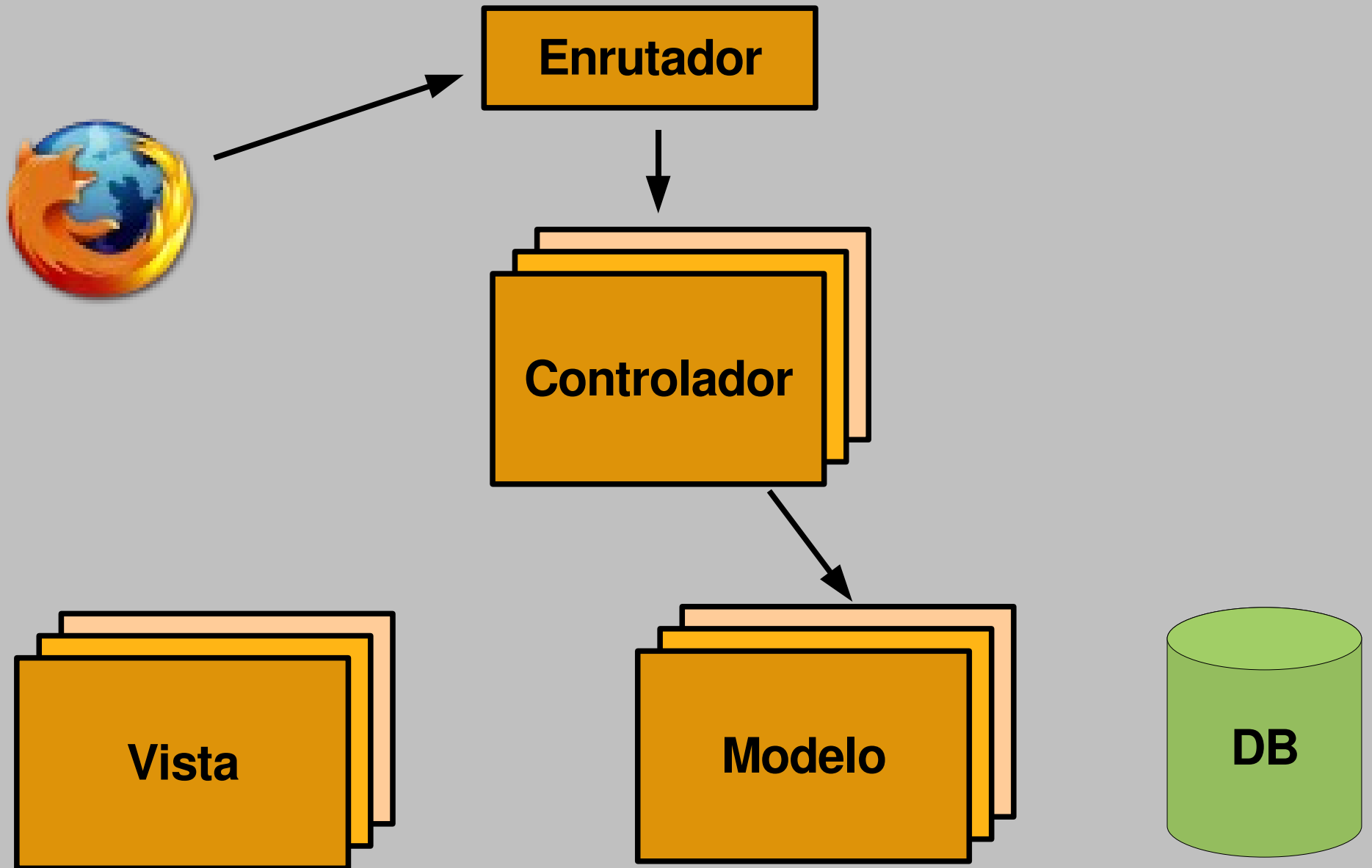
MVC en Rails



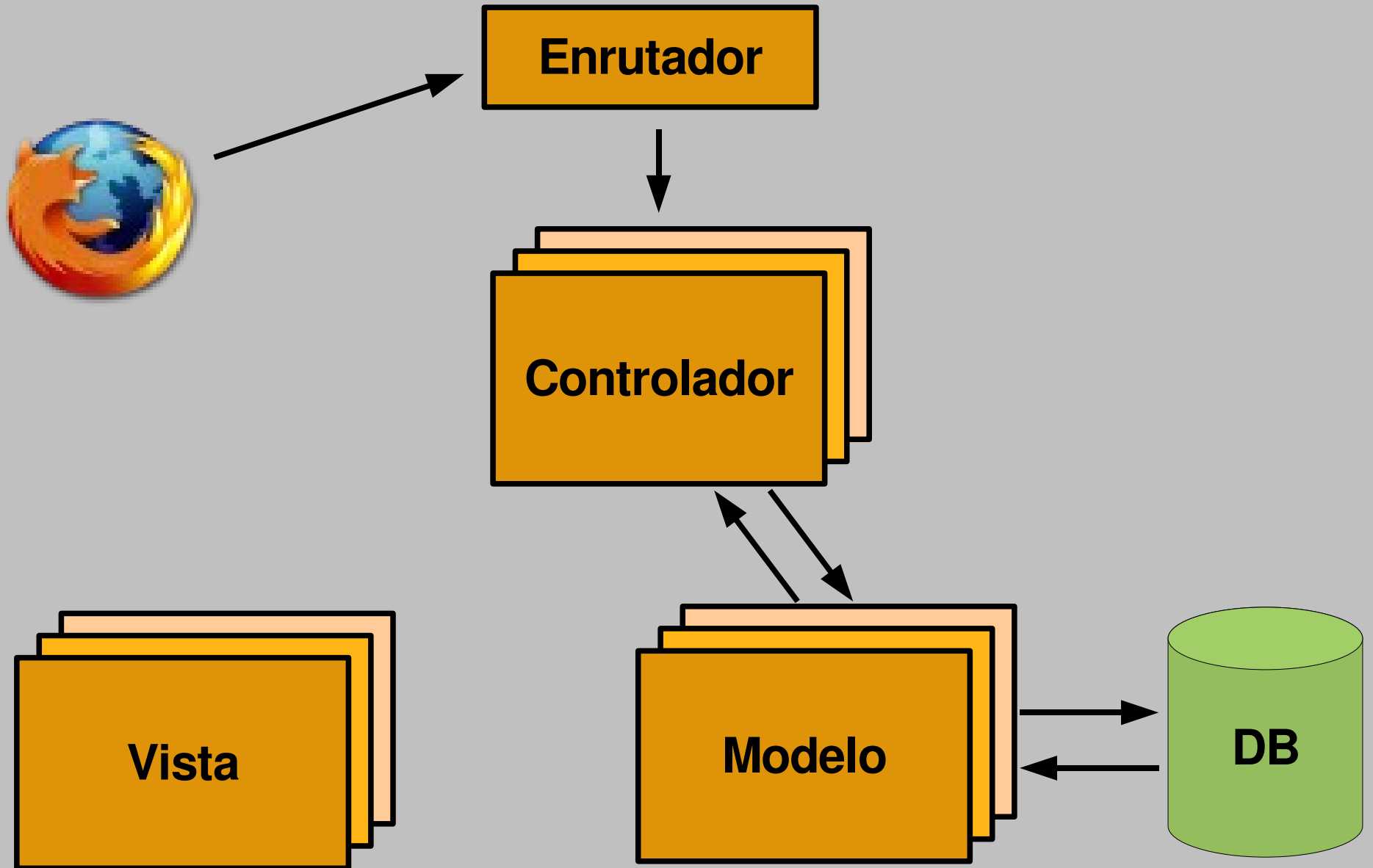
MVC en Rails



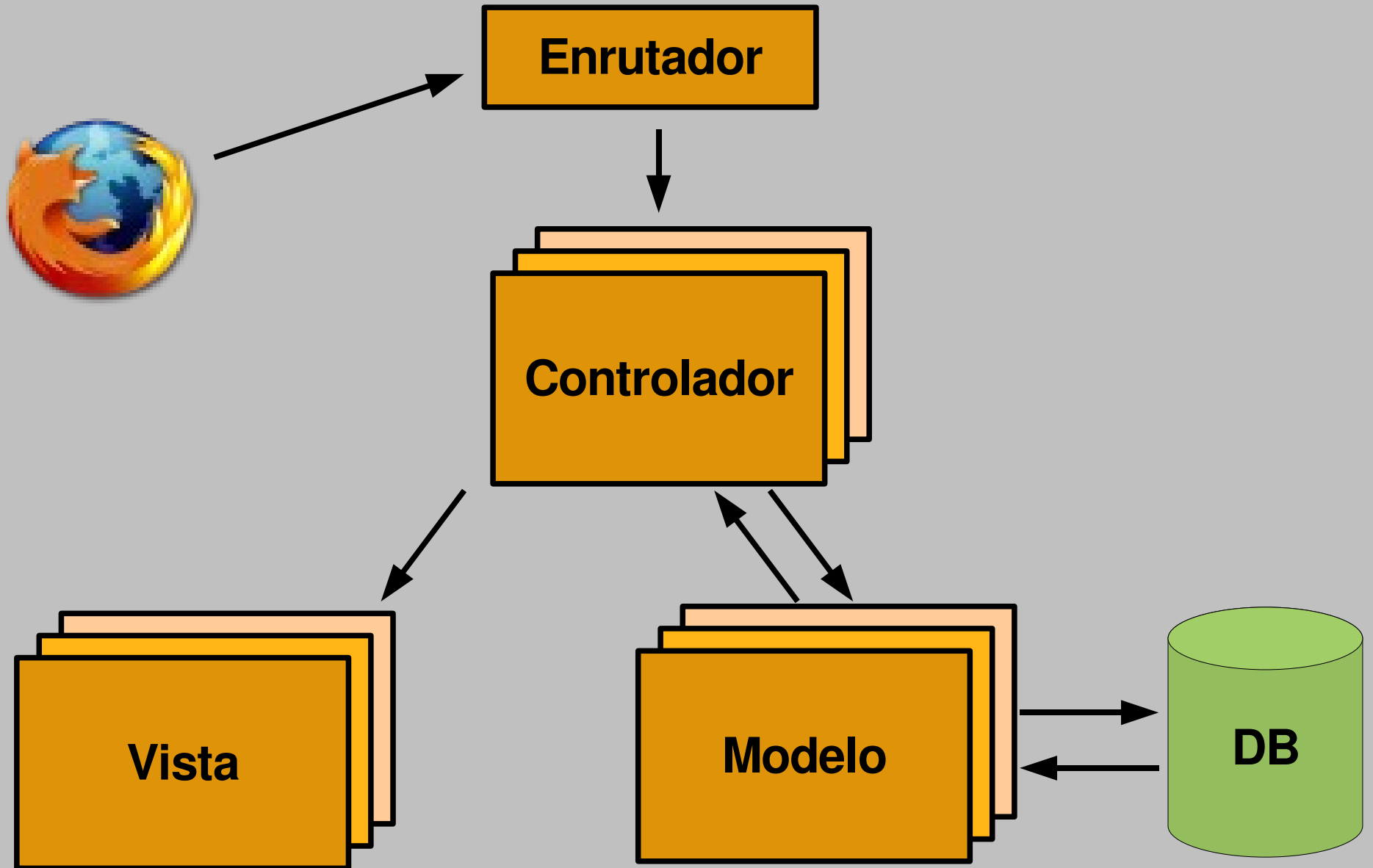
MVC en Rails



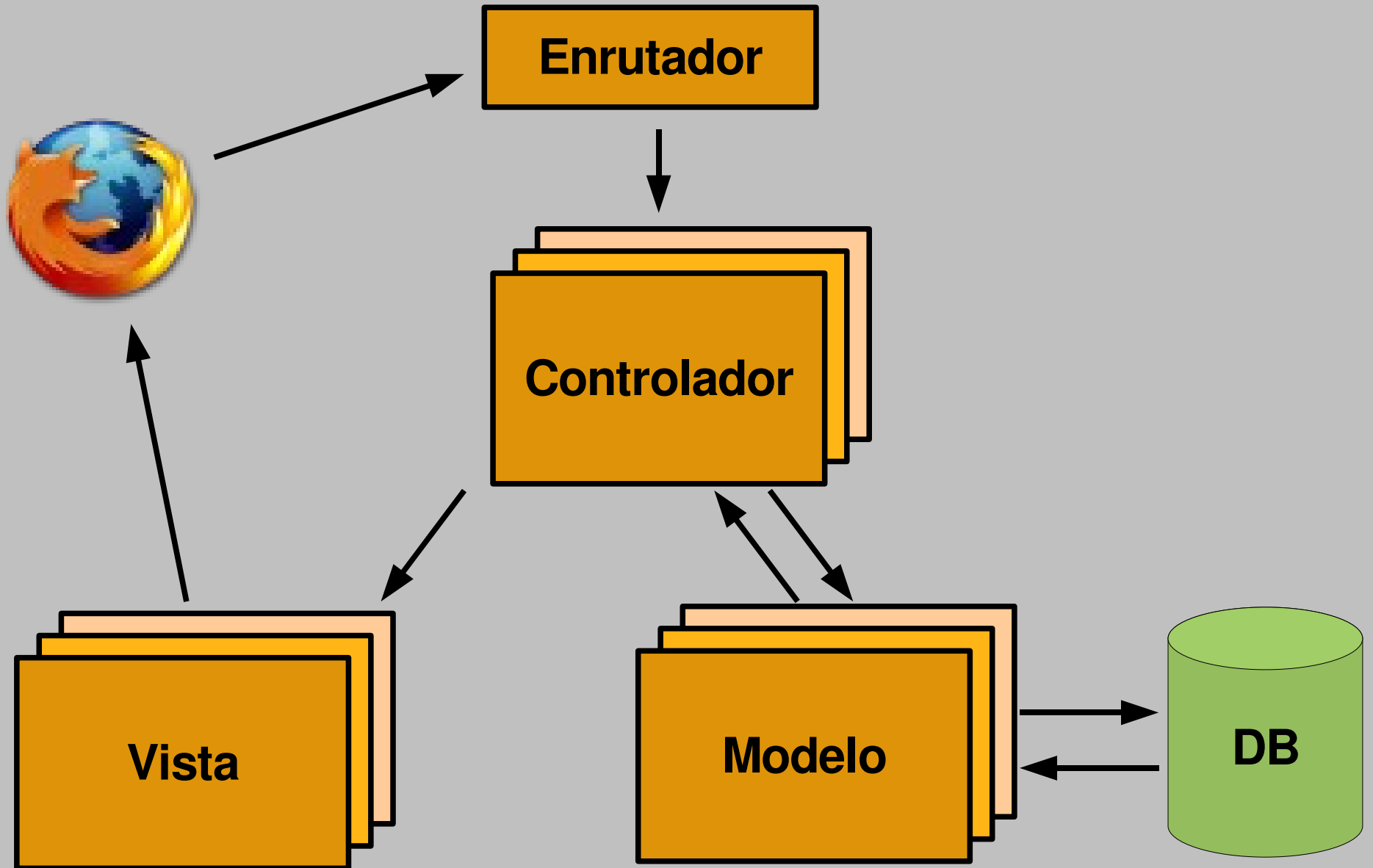
MVC en Rails



MVC en Rails



MVC en Rails



<http://server/controller/action/id>

<http://server/posts/show/1>

¿Qué es REST?

REST => “Representational State Transfer”

modelo de arquitectura de aplicaciones

- las aplicaciones gestionan **recursos**
- las operaciones son los **verbos HTTP**

REST => “Representational State Transfer”

modelo de arquitectura de aplicaciones

- las aplicaciones gestionan **recursos**
- las operaciones son los **verbos HTTP**



REST => “Representational State Transfer”

modelo de arquitectura de aplicaciones

- las aplicaciones gestionan **recursos**
- las operaciones son los **verbos HTTP**



Create

Read

Update

Delete

Create

Read

Update

Delete

insert

find

update

delete

Create

Read

Update

Delete

insert

find

update

delete

POST

GET

UPDATE

DELETE

Create

Read

Update

Delete

insert

find

update

delete

POST

GET

UPDATE

DELETE

create

show

update

destroy

POST **/posts**

GET **/posts/1**

PUT **/posts/1**

DELETE **/posts/1**

```
class PostsController < ApplicationController

  # GET /posts/1
  def show ... end

  # POST /posts
  def create ... end

  # PUT /posts/1
  def update ... end

  # DELETE /posts/1
  def destroy ... end

end
```

```
class PostsController < ApplicationController
```

```
# GET /posts/1  
def show ... end
```

```
# POST /posts  
def create ... end
```

```
# PUT /posts/1  
def update ... end
```

```
# DELETE /posts/1  
def destroy ... end
```

```
end
```

```
link_to "#{post.title}",  
post_url(post)
```

```
class PostsController < ApplicationController
```

```
  # GET /posts/1
```

```
  def show ... end
```

```
  # POST /posts
```

```
  def create ... end
```

```
  # PUT /posts/1
```

```
  def update ... end
```

```
  # DELETE /posts/1
```

```
  def destroy ... end
```

```
end
```

```
  form_for(:post,
```

```
    :url => posts_path) do |f|
```

```
    f.text_field :title
```

```
    f.text_area :body
```

```
    submit_tag "Create"
```

```
  end
```

```
class PostsController < ApplicationController
```

```
  # GET /posts/1
```

```
  def show ... end
```

```
  # POST /posts
```

```
  def create ... end
```

```
  # PUT /posts/1
```

```
  def update ... end
```

```
  # DELETE /posts/1
```

```
  def destroy ... end
```

```
end
```

```
  form_for(:post,
```

```
    :url => post_path(@post),
```

```
    :html => { :method => :put })
```

```
  do |f|
```

```
    f.text_field :title
```

```
    f.text_area :body
```

```
    submit_tag "Update"
```

```
  end
```

```
class PostsController < ApplicationController
```

```
  # GET /posts/1
```

```
  def show ... end
```

```
  # POST /posts
```

```
  def create ... end
```

```
  # PUT /posts/1
```

```
  def update ... end
```

```
  # DELETE /posts/1
```

```
  def destroy ... end
```

```
end
```

```
link_to 'Borrar',  
post_path(post),  
:method => :delete
```

Todavía hay más

```
class PostsController < ApplicationController
```

```
  # GET /posts
```

```
  def index ... end
```

```
  # GET /posts/1
```

```
  def show ... end
```

```
  # GET /posts/new
```

```
  def new ... end
```

```
  # GET /posts/1/edit
```

```
  def edit ... end
```

```
  # POST /posts
```

```
  def create ... end
```

```
  # PUT /posts/1
```

```
  def update ... end
```

```
  # DELETE /posts/1
```

```
  def destroy ... end
```

```
end
```

```
class PostsController < ApplicationController
```

```
# GET /posts  
def index ... end
```

```
# GET /posts/1  
def show ... end
```

```
# GET /posts/new  
def new ... end
```

```
# GET /posts/1/edit  
def edit ... end
```

```
# POST /posts  
def create ... end
```

```
# PUT /posts/1  
def update ... end
```

```
# DELETE /posts/1  
def destroy ... end
```

```
end
```

```
class PostsController < ApplicationController
```

```
  # GET /posts  
  def index ... end
```

```
  # GET /posts/1  
  def show ... end
```

```
  # GET /posts/new  
  def new ... end
```

```
  # GET /posts/1/edit  
  def edit ... end
```

```
  # POST /posts  
  def create ... end
```

```
  # PUT /posts/1  
  def update ... end
```

```
  # DELETE /posts/1  
  def destroy ... end
```

```
end
```

```
class PostsController < ApplicationController
```

```
  # GET /posts  
  def index ... end
```

```
  # GET /posts/1  
  def show ... end
```

```
  # GET /posts/new  
  def new ... end
```

```
  # GET /posts/1/edit  
  def edit ... end
```

```
  # POST /posts  
  def create ... end
```

```
  # PUT /posts/1  
  def update ... end
```

```
  # DELETE /posts/1  
  def destroy ... end
```

```
end
```

GET	/posts	index	posts_path
GET	/posts/1	show	post_path(1)
GET	/posts/new	new	new_post_path
GET	/posts/1/edit	edit	edit_post_path(1)
POST	/posts	create	posts_path
PUT	/posts/1	update	post_path(1)
DELETE	/posts/1	destroy	post_path(1)

```
class PostsController < ApplicationController
  def add_comment ... end
end
```

```
class PostsController < ApplicationController
  def add_comment ... end
end
```

```
class CommentsController < ApplicationController
  def assign_to_post ... end
end
```

```
class PostsController < ApplicationController
  def add_comment ... end
end
```

```
class CommentsController < ApplicationController
  def assign_to_post ... end
end
```

recursos anidados

```
class PostsController < ApplicationController
  def add_comment ... end
end
```

```
class CommentsController < ApplicationController
  def assign_to_post ... end
end
```

recursos anidados

añadiendo un nuevo recurso "membership"



```
class Post < ActiveRecord::Base
  has_many :memberships
  has_many :comments, :through => :memberships
end
```

```
class Comment < ActiveRecord::Base
  has_many :memberships
end
```

```
class Membership < ActiveRecord::Base
  belongs_to :comment
  belongs_to :post
end
```

Referencias

RailsConf '06: World of Resources

David Heinemeier Hansson

<http://media.rubyonrails.org/presentations/worldofresources.pdf>

Desarrollo REST con Rails

Ralf Wirdemann y Thomas Baustert

(traducción de Juan Lupión)

http://www.b-simple.de/download/restful_rails_es.pdf

RESTful Web Services

Leonard Richardson y Sam Ruby

<http://www.amazon.com/>

<http://javiervidal.net/>

hola@javiervidal.net