

P00, Módulos, Mixins y Rack _



**Rack es una herramienta para
construir aplicaciones web en ruby**

**Es utilizado como base por rails,
padrino y sinatra**

**Podemos trabajar en rails
sin saber mucho de rack**

¿Por qué estamos estudiando rack?

Por que nos ayudará a entender cómo funcionan los request y entender mejor cómo funciona rails

Instalemos rack

```
gem install rack
```

Una aplicación rack

- Tiene un objeto con el método call
- Acepta un parámetro con info del request
- La respuesta (como array)
- Con el código HTTP
- Headers (como el tipo de contenido)
- El Body

Nuestra primera app

```
#config.ru  
require 'rack'
```

```
class MiPrimeraWebApp
```

← 1) Objeto con el método call

```
  def call(env)
```

← 2) Info del request

```
    [200, {}, ['Hola']]
```

← 3) La respuesta

```
  end
```

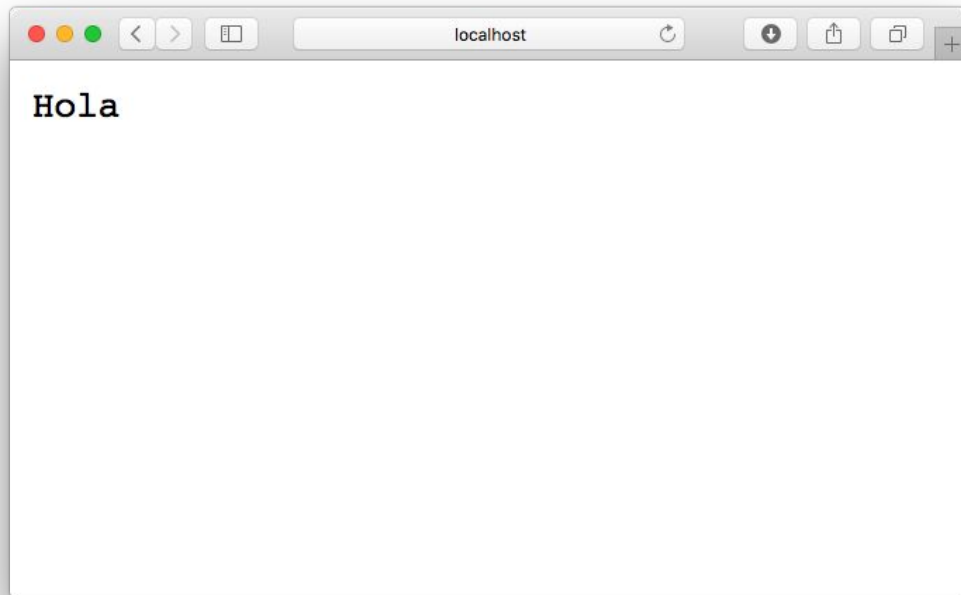
```
end
```

```
run MiPrimeraWebApp.new
```


Running rack

rackup

Localhost:9292



Info del request

```
{ "rack.version"=>[1, 3], "rack.errors"=>#<Rack::Lint::ErrorWrapper:0x007fb7b6961430 @error=#<IO:<STDERR>>>, "rack.multithread"=>true, "rack.multiprocess"=>false,
"rack.run_once"=>false, "SCRIPT_NAME"=>"", "QUERY_STRING"=>"", "SERVER_PROTOCOL"=>"HTTP/1.1", "SERVER_SOFTWARE"=>"puma 3.8.2 Sassy Salamander",
"GATEWAY_INTERFACE"=>"CGI/1.2", "REQUEST_METHOD"=>"GET", "REQUEST_PATH"=>"/", "REQUEST_URI"=>"/", "HTTP_VERSION"=>"HTTP/1.1", "HTTP_HOST"=>"localhost:9292",
"HTTP_ACCEPT"=>"text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8", "HTTP_UPGRADE_INSECURE_REQUESTS"=>"1",
"HTTP_COOKIE"=> "_ga=GA1.1.910282681.1488236145", "HTTP_USER_AGENT"=>"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_4) AppleWebKit/603.1.30 (KHTML, like Gecko)
Version/10.1 Safari/603.1.30", "HTTP_ACCEPT_LANGUAGE"=>"en-us", "HTTP_ACCEPT_ENCODING"=>"gzip, deflate", "HTTP_CONNECTION"=>"keep-alive",
"SERVER_NAME"=>"localhost", "SERVER_PORT"=>"9292", "PATH_INFO"=>"/", "REMOTE_ADDR"=> "::1", "puma.socket"=>#<TCPSocket:fd 15>, "rack.hijack?"=>true,
"rack.hijack"=>#<Proc:0x007fb7b69624c0@/Users/gonzalosanchez/.rvm/gems/ruby-2.3.1/gems/rack-2.0.1/lib/rack/lint.rb:525>,
"rack.input"=>#<Rack::Lint::InputWrapper:0x007fb7b69616d8 @input=#<Puma::NullIO:0x007fb7b5818818>>, "rack.url_scheme"=>"http", "rack.after_reply"=>[]],
"puma.config"=>#<Puma::Configuration:0x007fb7b69d0060 @options=#<Puma::UserFileDefaultOptions:0x007fb7b69c9a60 @user_options={:environment=>"development",
:pid=>nil, :Port=>9292, :Host=>"localhost", :AccessLog=>[] , :config=>"/Users/gonzalosanchez/Dropbox/Bootcamp fullstack B-learning/Experiencia 11/rack 1/config.ru",
:log_requests=>false, :binds=>["tcp://localhost:9292"] , :app=>#<Rack::ContentLength:0x007fb7b6275888 @app=#<Rack::Chunked:0x007fb7b62758d8
@app=#<Rack::CommonLogger:0x007fb7b6275928 @app=#<Rack::ShowExceptions:0x007fb7b627fd58 @app=#<Rack::Lint:0x007fb7b627fdd0
@app=#<Rack::TempfileReaper:0x007fb7b627fe48 @app=#<MiPrimeraWebApp:0x007fb7b627e758>>, @content_length=nil>>, @logger=#<IO:<STDERR>>>>>>, @file_options={},
@default_options={:min_threads=>0, :max_threads=>16, :log_requests=>false, :debug=>false, :binds=>["tcp://0.0.0.0:9292"] , :workers=>0, :daemon=>false, :mode=>:http,
:worker_timeout=>60, :worker_boot_timeout=>60, :worker_shutdown_timeout=>30, :remote_address=>:socket, :tag=>"rack 1", :environment=>"development", :rackup=>"config.ru",
:logger=>#<IO:<STDOUT>>>, :persistent_timeout=>20, :Verbose=>false, :Silent=>false}>, @plugins=#<Puma::PluginLoader:0x007fb7b69cb998 @instances=[]>,
@user_dsl=#<Puma::DSL:0x007fb7b69cb7e0 @config=#<Puma::Configuration:0x007fb7b69d0060 ...>, @options={:environment=>"development", :pid=>nil, :Port=>9292,
:Host=>"localhost", :AccessLog=>[] , :config=>"/Users/gonzalosanchez/Dropbox/Bootcamp fullstack B-learning/Experiencia 11/rack 1/config.ru", :log_requests=>false,
:binds=>["tcp://localhost:9292"] , :app=>#<Rack::ContentLength:0x007fb7b6275888 @app=#<Rack::Chunked:0x007fb7b62758d8
@app=#<Rack::CommonLogger:0x007fb7b6275928 @app=#<Rack::ShowExceptions:0x007fb7b627fd58 @app=#<Rack::Lint:0x007fb7b627fdd0
@app=#<Rack::TempfileReaper:0x007fb7b627fe48 @app=#<MiPrimeraWebApp:0x007fb7b627e758>>, @content_length=nil>>, @logger=#<IO:<STDERR>>>>>>, @plugins=[]>,
@file_dsl=#<Puma::DSL:0x007fb7b69cb538 @config=#<Puma::Configuration:0x007fb7b69d0060 ...>, @options={}, @plugins=[]>, @default_dsl=#<Puma::DSL:0x007fb7b69cb498
@config=#<Puma::Configuration:0x007fb7b69d0060 ...>, @options={:min_threads=>0, :max_threads=>16, :log_requests=>false, :debug=>false, :binds=>["tcp://0.0.0.0:9292"] ,
:workers=>0, :daemon=>false, :mode=>:http, :worker_timeout=>60, :worker_boot_timeout=>60, :worker_shutdown_timeout=>30, :remote_address=>:socket, :tag=>"rack 1",
:environment=>"development", :rackup=>"config.ru", :logger=>#<IO:<STDOUT>>>, :persistent_timeout=>20, :Verbose=>false, :Silent=>false}>, @plugins=[]>>, "rack.tempfiles"=>[]}]
:1 -- [11/May/2017:12:36:04 -0300] "GET / HTTP/1.1" 200 - 0.0013
{"rack.version"=>[1, 3], "rack.errors"=>#<Rack::Lint::ErrorWrapper:0x007fb7b70fcaf0 @error=#<IO:<STDERR>>>, "rack.multithread"=>true, "rack.multiprocess"=>false,
"rack.run_once"=>false, "SCRIPT_NAME"=>"", "QUERY_STRING"=>"", "SERVER_PROTOCOL"=>"HTTP/1.1", "SERVER_SOFTWARE"=>"puma 3.8.2 Sassy Salamander",
"GATEWAY_INTERFACE"=>"CGI/1.2", "REQUEST_METHOD"=>"GET", "REQUEST_PATH"=>"/", "REQUEST_URI"=>"/", "HTTP_VERSION"=>"HTTP/1.1", "HTTP_HOST"=>"localhost:9292",
"HTTP_ACCEPT"=>"text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8", "HTTP_UPGRADE_INSECURE_REQUESTS"=>"1",
"HTTP_COOKIE"=> "_ga=GA1.1.910282681.1488236145", "HTTP_USER_AGENT"=>"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_4) AppleWebKit/603.1.30 (KHTML, like Gecko)
```

La respuesta

```
[200, {}, ['Hola']]
```

El código http

```
[200, {}, ['Hola']]
```

100 => 'Continue',
101 => 'Switching Protocols',
102 => 'Processing',
200 => 'OK',
~~201 => 'Created',~~
202 => 'Accepted',
203 => 'Non-Authoritative Information',
204 => 'No Content',
205 => 'Reset Content',
206 => 'Partial Content',
207 => 'Multi-Status',
208 => 'Already Reported',
226 => 'IM Used',
300 => 'Multiple Choices',
301 => 'Moved Permanently',
302 => 'Found',
303 => 'See Other',
304 => 'Not Modified',
305 => 'Use Proxy',
307 => 'Temporary Redirect',
308 => 'Permanent Redirect',
400 => 'Bad Request',
401 => 'Unauthorized',
402 => 'Payment Required',
403 => 'Forbidden',
404 => 'Not Found',
405 => 'Method Not Allowed',
406 => 'Not Acceptable',
407 => 'Proxy Authentication Required',
408 => 'Request Timeout',
409 => 'Conflict',
410 => 'Gone',
411 => 'Length Required',
412 => 'Precondition Failed',
413 => 'Payload Too Large',
414 => 'URI Too Long',
415 => 'Unsupported Media Type',
416 => 'Range Not Satisfiable',
417 => 'Expectation Failed',
421 => 'Misdirected Request',
422 => 'Unprocessable Entity',
423 => 'Locked',
424 => 'Failed Dependency',
426 => 'Upgrade Required',
428 => 'Precondition Required',
429 => 'Too Many Requests',
431 => 'Request Header Fields Too Large',
451 => 'Unavailable for Legal Reasons',
500 => 'Internal Server Error',
501 => 'Not Implemented',
502 => 'Bad Gateway',
503 => 'Service Unavailable',

Response headers

Nos permiten especificar información adicional, como el tipo de contenido

```
[200, {}, ['Hola']]
```

Content type

```
[200, {'Content-Type' => 'text/html'}, ['<h1> Hola </h1>']]
```


Response body

Es el contenido entregado y que se mostrará al usuario

```
[200, {'Content-Type' => 'text/html'}, ['<h1> Hola </h1>']]
```

[\[200, {'Content-Type' => 'text/html'}, \['<h1> Hola </h1>'\]\]](#)

¿Y cómo podemos hacer que nuestra aplicación acepte más de una url?

- env guarda la información de la url pedida
- con `env['REQUEST_PATH']` podemos verla

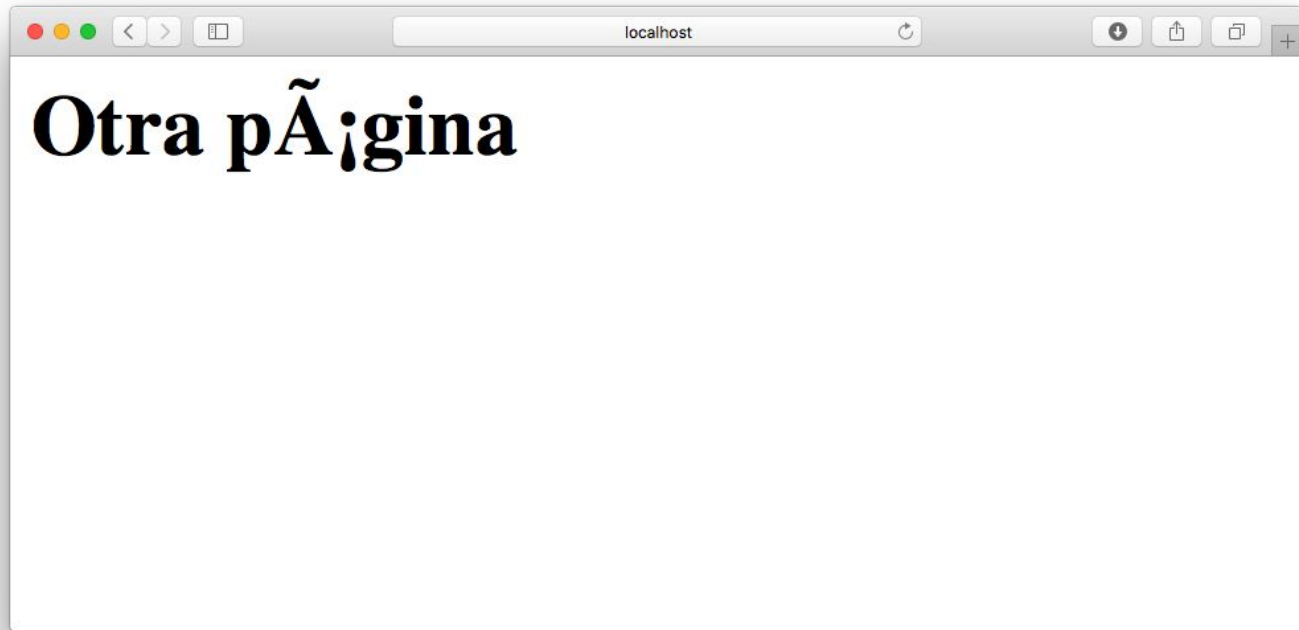
<http://localhost:9292/hola>



Luego si queremos manejar múltiples request, tenemos que utilizar un if (o un case)

```
#config.ru
require 'rack'
class MiPrimeraWebApp
  def call(env)
    if env['REQUEST_PATH'] == '/hola'
      [200, {'Content-Type' => 'text/html'}, ["<h1> Página 1 </h1>"]]
    else
      [200, {'Content-Type' => 'text/html'}, ["<h1> Otra página </h1>"]]
    end
  end
end

run MiPrimeraWebApp.new
```



Modificando el código sin reiniciar rack

```
use Rack::Reloader, 0
```

Cargando archivos html

```
#config.ru
require 'rack'

class MiPrimeraWebApp
  def call(env)
    if env['REQUEST_PATH'] == '/hola'
      [200, {'Content-Type' => 'text/html'}, ["<h1> Página 1
</h1>"]]
    else
      [200, {'Content-Type' => 'text/html'},
      [File.read("index.html")]]
    end
  end
end
run MiPrimeraWebApp.new
```

Profundizar

<https://www.youtube.com/watch?v=NJ-ilQMsqMs>

