

# Tareas en background con RoR y BackgroundDRb



IULA – UPF

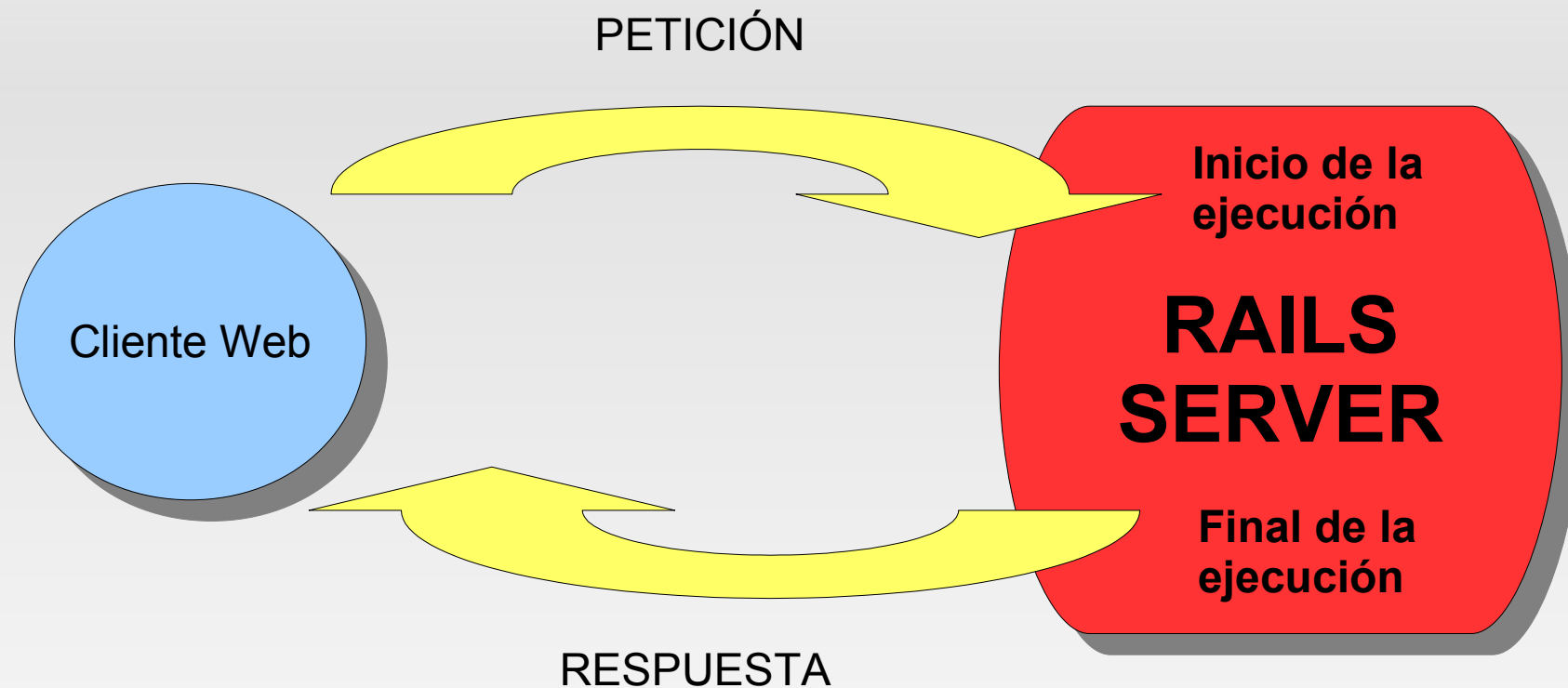
Este trabajo se enmarca dentro del proyecto AAILE (financiado por el Ministerio de Educación y Ciencia y el Fondo Social Europeo PTA-CTE/1370/2003)

[sergio.espeja@upf.edu](mailto:sergio.espeja@upf.edu)

# ¿ Te gusta esperar ?



# Ciclo petición/respuesta



# Problema

- Procesos de cálculo largos
- Actualizaciones masivas de datos
- Peticiones de datos a servidores externos:
  - WebServices
  - Transferencia de archivos
- etc ...

# Soluciones posibles

- UNIX Cron y script/runner
  - script/runner “Modelo.accion”
    - Estable pero consume muchos recursos
    - No ejecutable desde la aplicación rails
- Forks o Threads
  - Programación a muy bajo nivel
  - Fork da problemas en Windows
  - Threads complican la instalación de servidores web como Apache Httpd

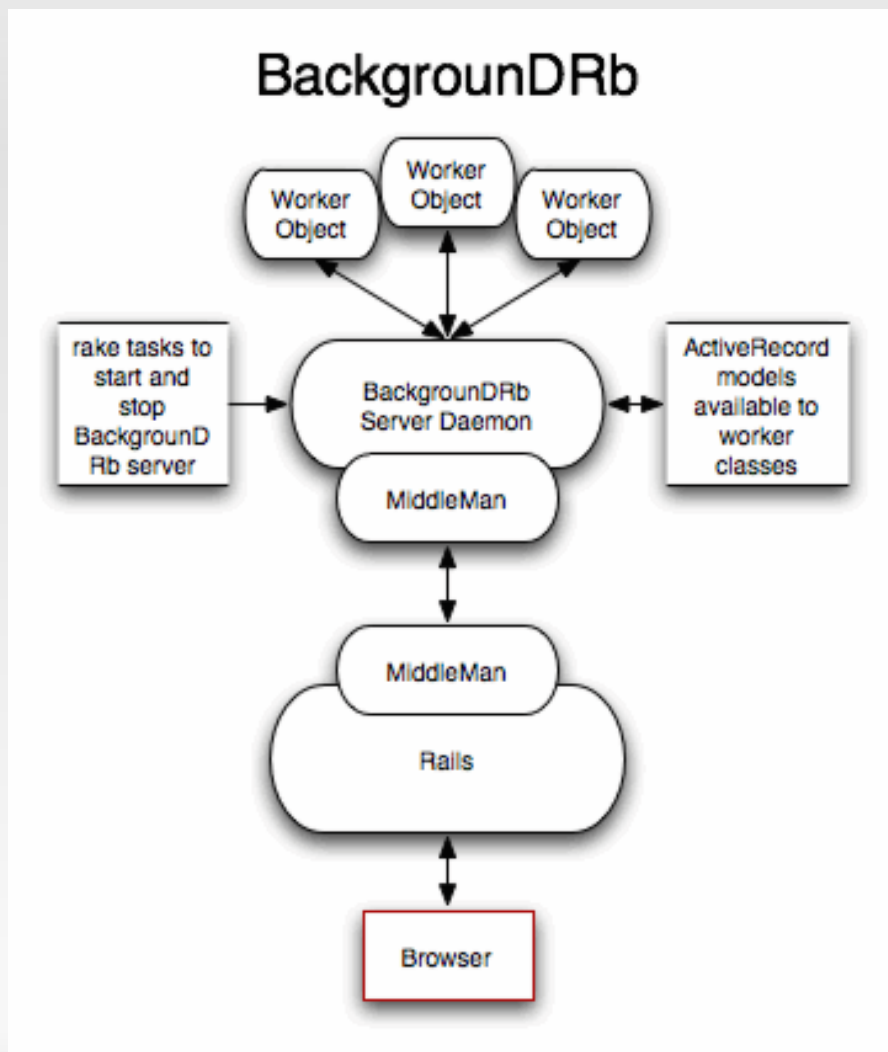
# BackgroundRb

- Framework para extraer tareas del ciclo de petición/respuesta en RoR
- Permite fácilmente la comunicación entre la tarea y la aplicación rails
- Servidor de objetos (DRb)
  - <http://chadfowler.com/ruby/drb.html>
- Desarrollado por:
  - Ezra Zygmuntowicz (<http://brainspl.at/>)
  - [skaar@waste.org](mailto:skaar@waste.org)

# BackgroundRb, posibles aplicaciones

- Caching
  - Mantener en cache objetos ActiveRecord
- Mantener sesiones remotas abiertas
  - Cliente web de servicios de mensajería instantánea (ej. Jabber)
- Servidor de tareas en background

# Arquitectura BackgroundDRb



- Servidor de tareas
  - Workers
  - MiddleMan
- Aplicación Rails
  - MiddleMan



¡Creación de un contador de segundos!

# Instalación

- `script/plugin install`  
`svn://rubyforge.org/var/svn/backgroundrb`

`vendor/plugins/backgroundrb/backgroundrb_rails.rb`

`vendor/plugins/backgroundrb/backgroundrb.rb`

`vendor/plugins/backgroundrb/init.rb`

`vendor/plugins/backgroundrb/lib/backgroundrb.rb`

`vendor/plugins/backgroundrb/tasks/backgroundrb_tasks.rake`

`vendor/plugins/backgroundrb/script/backgroundrb/stop`

`vendor/plugins/backgroundrb/script/backgroundrb/start`

`vendor/plugins/backgroundrb/generators/worker/worker_generator.rb`

`vendor/plugins/backgroundrb/examples`

`vendor/plugins/backgroundrb/test`

- `rake backgroundrb:setup`

`config/backgroundrb.yml`

`script/background`

`lib/workers`

# config/backgroundrb.yml

---

**port:** "22222"

**timer\_sleep:** 60

**load\_rails:** true

**environment:** development

**host:** localhost

**database\_yml:** config/database.yml

**acl:**

deny: all

allow: localhost 127.0.0.1

order: deny,allow

# Puesta en marcha

- Encender:
  - `ruby script/backgroundrb/start`
- Apagar:
  - `ruby script/backgroundrb/stop`

# Creación de un worker

- `script/generate worker Currante`
  - `lib/workers/currante_worker.rb`
  - `test/unit/currante_worker_test.rb`

# lib/workers/currante\_worker.rb

```
class CurranteWorker < BackgroundDRb::Rails

  attr_reader :progreso, :total, :resultado

  def do_work(args)
    @progreso = 0; @total = args
    @total.times {
      sleep 1
      @progreso += 1
    }
    @resultado = "He contado #{args} segundos!"
    ActiveRecord::Base.connection.disconnect!
  end

end
```

# lib/workers/currante\_worker.rb

```
class CurranteWorker < BackgroundDRb::Rails

  attr_reader :progreso, :total, :resultado

  def do_work(args)
    @progreso = 0; @total = args
    @total.times {
      sleep 1
      @progreso += 1
    }
    @resultado = "He contado #{args} segundos!"

    ActiveRecord::Base.connection.disconnect!

  end

end
```

# Poner a trabajar a nuestro worker

- `script/generate controller Vigilante`
  - `app/controllers/vigilante_controller.rb`
  - `test/functional/vigilante_controller_test.rb`
  - `app/helpers/vigilante_helper.rb`



# app/controllers/vigilante\_controller.rb

```
class VigilanteController < ApplicationController

  def index
  end

  def lanzar_tarea
    # Crear el worker
    session[:jobkey_currante] = MiddleMan.new_worker :class => :currante_worker,
                                                    :args => 30
  end

  def progreso
    # Actualizar el % finalizado
    if session[:jobkey_currante]
      @progreso = MiddleMan.get_worker(session[:jobkey_currante]).progreso
      @total = MiddleMan.get_worker(session[:jobkey_currante]).total
      @completado = @progreso.to_f * 100.0 / @total.to_f
      redirect_to :action => "finalizado" if @progreso >= @total
    end
  end

  def finalizado
    # Mostrar resultados
    @resultado = MiddleMan.get_worker(session[:jobkey_currante]).resultado
    # Eliminar al worker
    MiddleMan.delete_worker(session[:jobkey_currante])
    session[:jobkey_currante] = nil
  end

end
```

# app/views/vigilante/index.rhtml

```
<html>
<head>
  <meta http-equiv="Content-type" content="text/html; charset=utf-8">
  <title>Conferencia Rails Hispana: BackgroundDRb </title>
  <%= javascript_include_tag :defaults %>
</head>
<body bgcolor="#F8FAEC" style="font-size: 1.5em; font-family: georgia;">
<center>

  <div style="text-align: center; margin-top:100px; padding: 10px; width:70%;
            background-color:#ccc; border:1px solid;">

    <%= periodically_call_remote :url => { :action => "progreso" },
                                :frequency => 1 %>

    <div id="progreso"></div>

    <div id="tarea">
      <%= link_to_remote "Hacer currar", :url => { :action => "lanzar_tarea" } %>
    </div>

  </div>

</center>
</body>
</html>
```

# app/views/vigilante/\*.rjs

## app/views/vigilante/lanzar\_tarea.rjs:

```
if session[:jobkey_currante]
  page.replace_html 'tarea', "Currante con job key #{session[:jobkey_currante]} creado"
end
```

## app/views/vigilante/progreso.rjs:

```
if @completado
  page.replace_html 'progreso', " #{ "%.2f" % @completado } % Completado"
end
```

## app/views/vigilante/finalizado.rjs:

```
page.replace_html 'progreso', "100.00 % Finalizado"
page.replace_html 'tarea', "Resultado: <pre>#{@resultado}</pre> " \
  + link_to_remote("Hacer currar de nuevo!", :url => {:action => "lanzar_tarea" })
```

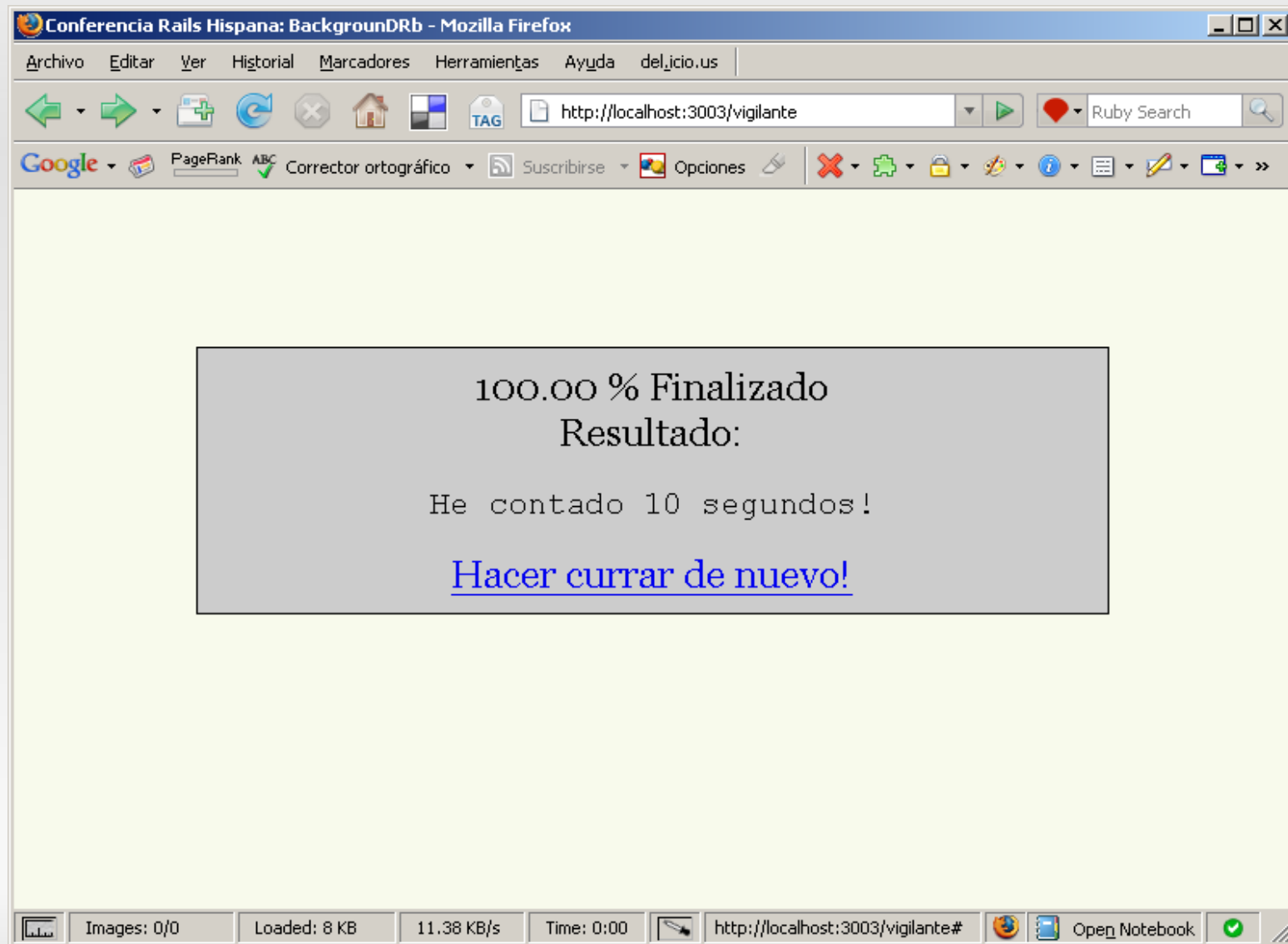
# Resultado

The screenshot shows a Mozilla Firefox browser window with the title "Conferencia Rails Hispana: BackgroundDRb - Mozilla Firefox". The address bar contains "http://localhost:3003/vigilante". The main content area displays a grey box with the following text:

36.67 % Completado  
Currante con job key  
fa7137a7584dd83c9ba66d3b0c619607 creado

The browser's status bar at the bottom shows "Images: 0/0", "Loaded: 8 KB", "11.38 KB/s", "Time: 0:00", and "Terminado".

# Resultado



# BackgroundRb en desarrollo

- Versión < 0.2.0 (utilizada en los ejemplos)
  - Instalable como plugin
  - 1 proceso, multi threaded
- Versión 0.2.0 (actual)
  - **Experimental**, instalable directamente del svn
  - Multiproceso y más robusta
- Versión 0.3.x (futura)
  - Apta para producción

# Más información

- BackgroundDRB version < 0.2.0
  - vendor/plugins/backgroundrb/README
- BackgroundDRB version 0.2.0
  - <http://backgroundrb.rubyforge.org>
  - <http://brainspl.at/articles/tag/background>
  - <http://backgroundrb.devjavu.com>
- Transparencias + Código + Recursos
  - <http://bee.com.es/espeja/conferenciarails>

Gracias!

**¿PREGUNTAS?**