# Tutorial: Servidor Rest utilizando Ruby

## Elaboración Servicio Web Rest

A continuación se detallan los pasos para el desarrollo de un Servicio Web Rest mediante Ruby.

## Herramientas a instalar

Para poder realizar servicios web utilizando Ruby, es necesario tener instalados las siguientes herramientas:

* Ruby
* Rails

### Instalar Ruby en Windows

La instalación de Ruby en Windows se puede realizar de dos formas, utilizando Ruby Versión Manager (RVM) o un instalador (.exe).

En este tutorial se utilizará RubyInstaller porque la herramienta RVM dejó de tener soporte en 2012. Para descargar RubyInstaller hay que ir a la página <http://rubyinstaller.org/downloads/> y seleccionar el instalador Ruby 2.1.5 o Ruby 2.1.5(x64).

Para ver que Ruby ha sido instalado correctamente y que la versión es 2.5.1 es suficiente con escribir ruby –version en la línea de comandos.

ruby --version

### Instalar Ruby en UNIX

Para la instalación de Ruby en sistemas operativos UNIX se debe utilizar la herramienta Ruby Versión Manager (RVM). El comando para instalar RVM y la última versión estable de Ruby es:

\curl -L https://get.rvm.io | bash -s stable --ruby

### Instalar Rails (Windows y UNIX)

Rails es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby. Para instalar Rails en Windows o en cualquier sistema operativa UNIX, el comando es el mismo. Lo único que cambia es que en UNIX se debe de poner siempre la palabra sudo antes del comando. El comando que se debe escribir en la línea de comandos es:

Windows

gem install rails

UNIX

sudo gem install rails

### Problemas instalando la gema

Al intentar instalar gemas a veces se obtiene el siguiente problema “*Unable to download data from https://rubygems.org/ - SSL\_connect returned=1 errno=0 state=SSLv3 read server certificate B: certificate verify failed*”. Para solucionar este problema, se debe ejecutar en la línea de comandos:

Windows

gem sources –r <https://rubygems.org>

gem sources –a <http://rubygems.org>

UNIX

sudo gem sources –r <https://rubygems.org>

sudo gem sources –a <http://rubygems.org>

## Creación del proyecto

1. Crear directorio “*TutorialRuby*” en C. Sí ya existe, simplemenet acceder a él.
2. Abrir la línea de comandos e ir al directorio donde se encuentra “*TutorialRuby*”.

cd C:\TutorialRuby

1. Crear un nuevo proyecto Rails utilizando el comando “*rails new nombreproyecto*”. En este caso el nombre del proyecto será “*RestServer*”.

rails new RestServer

1. Una vez terminada la creación del proyecto, deberá tener los siguientes directorios en “*C:\TutorialRuby\RestServer*”.

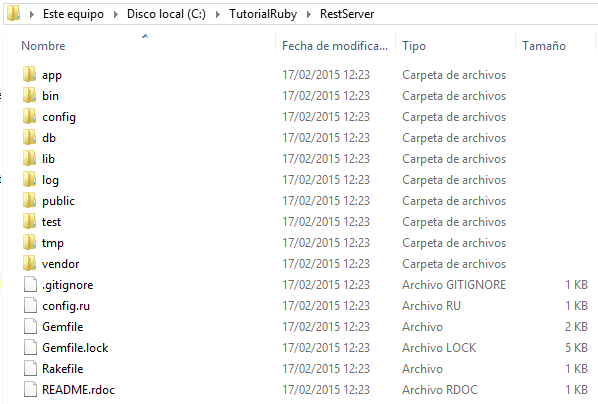


Ilustración : Directorios del proyecto RestServer

## Implementación del Servicio Calculator

1. Acceder al directorio “*RestServer/app/controllers*” mediante la línea de comandos.

cd C:\TutorialRuby\RestServer\app\controllers

1. Crear el controlador calculator con el comando

rails generate controller calculator\_rest add

1. Abrir el fichero calculator\_rest\_controller con un editor de texto y copiar el código.

class CalculatorRestController < ApplicationController

def add

render :text => params['a'].to\_i + params['b'].to\_i

end

end

1. Buscar el fichero “*routes*” dentro del directorio “*RestServer/config*” y copiar las siguientes líneas.

get 'welcome/index'

get 'calculator\_rest/' => 'calculator\_rest#index'

get 'calculator\_rest/add/:a/:b' => 'calculator\_rest#add'

1. Acceder al directorio “*bin*” mediante línea de comando y arrancar el servidor.

cd C:\TutorialRuby\RestServer\bin

rails server

1. Para verificar que todo funciona correctamente, copiar en el navegador la url “[*http://localhost:3000/calculator\_rest/add*](http://localhost:3000/calculator_rest/add)”.

## Creación del cliente en Ruby

1. Crear y acceder al directorio “*RestClient*” dentro de “*TutorialRuby*”.
2. Abrir un editor de texto, copiar el código y guárdalo con el nombre “*rest\_client.rb*”. Revisar que no guarda con la extensión .txt.

require 'net/http'

class RestClient

#http://localhost/calculator/rest

#/operation/2/2

def initialize(urlBase, operation, term1, term2)

@url=urlBase

@operation=operation

@term1=term1

@term2=term2

case @operation

when "suma"

puts sumar(@term1,@term2)

else

puts "La operacion no es correcta, elige: suma"

end

end

def sumar(term1, term2)

@url+="add/"+term1+"/"+term2

resp=Net::HTTP.get\_response(URI.parse(@url))

resp.body

end

end

urlBase="http://localhost:3000/calculator\_rest/"

operation=ARGV[0]

term1=ARGV[1]

term2=ARGV[2]

restClient=RestClient.new(urlBase, operation, term1, term2)

1. Para probar que funciona correctamente, acceder mediante línea de comando al directorio “*TutorialRuby/RestClient*” y ejecutar el comando “ruby nombreficherocliente.rb operación term1 term2”. Ejemplo para realizar la suma:

cd C:\TutorialRuby\RestClient

ruby rest\_client.rb suma 1 1