

RUBY ON RAILS

Desarrollo práctico
de aplicaciones web

Santiago Ponce Moreno



Ruby on Rails. Desarrollo práctico de aplicaciones web
Santiago Ponce Moreno

ISBN: 978-84-941272-7-4

EAN: 9788494127274

IBIC: UMW

Copyright © 2013 RC Libros

© RC Libros es un sello y marca comercial registrados

Ruby on Rails. Desarrollo práctico de aplicaciones web

Reservados todos los derechos. Ninguna parte de este libro incluida la cubierta puede ser reproducida, su contenido está protegido por la Ley vigente que establece penas de prisión y/o multas a quienes intencionadamente reprodujeran o plagiaran, en todo o en parte, una obra literaria, artística o científica, o su transformación, interpretación o ejecución en cualquier tipo de soporte existente o de próxima invención, sin autorización previa y por escrito de los titulares de los derechos del copyright. La infracción de los derechos citados puede constituir delito contra la propiedad intelectual. (Art. 270 y siguientes del Código Penal). Dirijase a CEDRO (Centro Español de Derechos Reprográficos) si necesita fotocopiar o escanear algún fragmento de esta obra a través de la web www.conlicencia.com; o por teléfono a: 91 702 19 70 / 93 272 04 47)

RC Libros, el Autor, y cualquier persona o empresa participante en la redacción, edición o producción de este libro, en ningún caso serán responsables de los resultados del uso de su contenido, ni de cualquier violación de patentes o derechos de terceras partes. El objetivo de la obra es proporcionar al lector conocimientos precisos y acreditados sobre el tema tratado pero su venta no supone ninguna forma de asistencia legal, administrativa ni de ningún otro tipo, si se precisase ayuda adicional o experta deberán buscarse los servicios de profesionales competentes. Productos y marcas citados en su contenido estén o no registrados, pertenecen a sus respectivos propietarios.

RC Libros

Calle Mar Mediterráneo, 2. Nave 6
28830 SAN FERNANDO DE HENARES, Madrid

Teléfono: +34 91 677 57 22

Fax: +34 91 677 57 22

Correo electrónico: info@rclibros.es

Internet: www.rclibros.es

Diseño de colección, cubierta y pre-impresión: Grupo RC

Impresión y encuadernación: Service Point

Depósito Legal: M-24937-2013

Impreso en España

17 16 15 14 13 (1 2 3 4 5 6 7 8 9 10 11 12)

PARTE I

CURSO PRÁCTICO DE RUBY

1. INSTALACIÓN DE RUBY ON RAILS
2. LENGUAJE RUBY
3. MÉTODOS, BLOQUES, SÍMBOLOS Y EXPRESIONES REGULARES
4. CLASES
5. HERENCIA
6. EXCEPCIONES, DEPURACIÓN Y PRUEBAS UNITARIAS
7. API RUBY

1 INSTALACIÓN DE RUBY ON RAILS

INTRODUCCIÓN

El primer paso como suele ser habitual en estas ocasiones cuando empezamos con una nueva tecnología es la instalación de todas las herramientas necesarias para poder empezar a programar. Hay que tener en cuenta una serie de consideraciones previas a la hora de poder ejecutar Ruby on Rails (RoR) en nuestro sistema operativo. Para poder realizar dicha acción con éxito, es preciso tener instalados los siguientes componentes:

- Un **intérprete de Ruby (IRB)**. Es el software necesario para que pueda interpretar el código del lenguaje de programación orientado a objetos Ruby y ejecutarlo. La versión que utilizaremos para los ejemplos es la 1.9.3.
- No podemos olvidarnos del framework **Rails** que lo instalaremos como una gema en nuestro sistema. En lo que nos atañe, nos atenderemos a usar la versión 3.2.9.
- Un **servidor web**. Para poder ver nuestras aplicaciones web emplearemos un servidor web que lanzaremos en local, llamado *WEBrick*, es el que se instala por defecto junto con *Rails*. Dentro del mismo disponemos de tres ambientes de trabajo: test, desarrollo y producción. Cada uno de ellos con su propia configuración para mejorar el rendimiento según su cometido.

- Ambiente de trabajo **Test**, su fin es chequear nuestra aplicación; por tanto, los datos almacenados en la base de datos son destruidos cada vez que reiniciamos el servidor, por ello hay que tener especial cuidado y por seguridad tenemos que utilizar diferentes nombres de base de datos, cada uno de ellos para un ambiente determinado, es aconsejable seguir esta nomenclatura.

nombre_base_de_datos_<ambiente de trabajo>

- Ambiente de trabajo **Desarrollo**, es recomendable utilizarlo cuando nos encontramos desarrollando nuestra aplicación. En cada petición cargará de nuevo todo nuestro código fuente, interferirá en un menor rendimiento del servidor, pero se reducirá considerablemente la necesidad de reiniciar el mismo.
- Ambiente de **Producción**, es lanzado cuando damos por finalizada nuestra aplicación, por tanto está configurado para que no muestre las excepciones que pueda provocar nuestro código, eliminando la necesidad de cargar todo en cada petición para mejorar el rendimiento. Se supone que nuestra aplicación ya es pública para que los usuarios puedan interactuar con ella y nos veremos en la obligación de implementar un servidor web más avanzado y con mejores características. De momento, en el punto donde nos encontramos nos bastaría con utilizar el servidor *WEBrick*.
- Necesitamos el software necesario para escribir nuestro código, las posibilidades son utilizar un **editor** o un **IDE (Integrated Development Environment)**. Aquí conoceremos el editor denominado *Sublime Text2* y sus características, aunque hay multitudes de opciones totalmente válidas. Este editor no es gratuito, pero su uso con el objeto de ser empleado didácticamente, no plantea limitaciones. Por otro lado, seguiremos empleando este gran editor, por su coste, sus ventajas y porque resulta muy asequible.

En primer lugar detallaremos la instalación de Ruby on Rails en los diferentes sistemas operativos, y acto seguido el editor seleccionado por el autor para el seguimiento del libro.

Instalación en Windows

Los usuarios de Windows, en este caso en concreto, están de enhorabuena. La empresa *Engine Yard* pone a disposición un pack que permite instalar todo lo necesario para comenzar sin mas dilaciones. Su nombre es *RailsInstaller* y lo podemos encontrar en la página web <http://www.railsinstaller.org>.



Fig. 1-1 Página web de RailsInstaller

Este pack incluye los siguientes componentes destacables:

- La instalación de los anteriormente citados por imprescindibles, **Ruby** y la gema **Rails**.
- **Bundler**. Es un sistema de control de dependencias sobre las gemas. Es el responsable de tener constancia de qué plugins se han instalado y qué dependencias tienen para instalarlas y conseguir su correcto funcionamiento. En los proyectos que realicemos existe un archivo en el directorio raíz llamado *Gemfile.lock* donde se registran las versiones instaladas de las gemas, facilitando la exportación a otra máquina.
- **Git**. Es un software de control de versiones de nuestro código fuente diseñado por *Linus Torvalds*. A medida que nuestra aplicación web aumenta en funcionalidades vamos documentándolo con repositorios GIT, con la ventaja que si alguna actualización pudiera provocar cierta inestabilidad al programa web, GIT nos permitiría volver a una versión anterior, además de tener

documentadas todas las ampliaciones con la funcionalidad que se le ha implementado. Está basada en repositorios locales y admite la posibilidad de trabajar en equipo. Su complemento ideal es *GitHub*, que facilita la opción de tener nuestros archivos en la *nube* de forma pública y visual, realizando una subida desde el repositorio local GIT a *GitHub*. Tenemos la posibilidad de tener una cuenta privada de pago.

- **SQLite.** Es una librería escrita en C que implementa un motor de base de datos. Se diferencia significativamente de otras, ya que está basada en un servicio externo, esta se integra completamente con nuestra aplicación e incluso viene configurada en los nuevos proyectos generados de *Rails* para poder comenzar a trabajar con bases de datos sin tener que realizar ninguna operación adicional. Eso sí, tiene algunas limitaciones como la de no permitir acceder a múltiples usuarios en modo escritura, por lo que es frecuente que se utilice otro tipo de base de datos para el entorno de producción. En la versión 3 permite hasta 2 terabytes de tamaño en su base de datos y será la que emplearemos cuando estemos en fase de desarrollo.

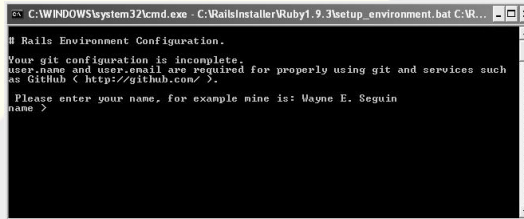
Has de conocer que, además de estos componentes, también se instala en el mismo proceso, *TinyTDS* (https://github.com/rails-sqlserver/tiny_tds), *SQL Server Support* (<https://github.com/rails-sqlserver/activerecord-sqlserver-adapter>) y *DevKit* (<https://github.com/oneclick/rubyinstaller/wiki/Development-Kit>), cuya importancia es relativa, y es por ello que no los explicamos, pero si tiene curiosidad podemos acudir a su web correspondiente para documentarse. Centrémonos en conocer cómo instalar RoR en Windows, y para ello vamos a describir los siguientes pasos.

1. Realizaremos la instalación del **pack *RailsInstaller***, siguiendo las opciones que, por defecto, vienen marcadas.



Fig. 1-2 RailsInstaller

2. Terminada la instalación con éxito, veremos cómo se abre una consola donde introduciremos el usuario y el email de nuestra cuenta *GitHub*. Y, aunque hasta que lleguemos a niveles más avanzados no será objeto de tratamiento su modo empleo, vamos a crearla y configurarla.



```

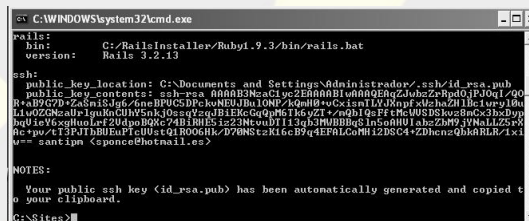
C:\WINDOWS\system32\cmd.exe - C:\RailsInstaller\Ruby1.9.3\setup_environment.bat C:\R...
# Rails: Environment Configuration.
Your git configuration is incomplete.
user_name and user_email are required for properly using git and services such
as GitHub < http://github.com >.

Please enter your name, for example mine is: Wayne E. Seguin
name >

```

Fig. 1-3 Datos GitHub

- a. Es entonces cuando nos dirigimos a la página web <http://www.github.com> y sin más creamos una nueva cuenta.
- b. Una vez realizados todos los pasos citados anteriormente sin incidencias, introduciremos los datos solicitados en la consola CMD de Windows.



```

C:\WINDOWS\system32\cmd.exe
rails:
bin: C:\RailsInstaller\Ruby1.9.3\bin\rails.bat
version: Rails 3.2.13

ssh:
public_key_location: C:\Documents and Settings\Administrador/.ssh/id_rsa.pub
public_key_contents: ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAqZJubhzZcRpd0jF10qL90
R+AD9C7D*Za8n1Sj67neBPUC5DpkoNEUJbUIONP*kdqH0+Cs3nILD4npt4Ztzh2H1bc1uey10a
L140ZC8e3a1e1pubkHW5n3j5e3Yed11EicqQm8Tksytr*wdB1E3F3e3u3S3u3h3w333d3y
bqVieV6xghuolwF2U4p0B9Kc74B3H8E51z23Nt5ud113q3MUBB8q61n5o8HU1abzZM93VMeLLZ5x8
0c*pu*413P11B8B8Tf0U8tq1B066Rk/D70NStzK16cB9q4FALCoM12DS4C4ZDhczQ6kRLR/1x1
e= santia @pence@hotmail.es

NOTES:
Your public ssh key <id_rsa.pub> has been automatically generated and copied t
o your clipboard.
C:\$>

```

Fig. 1-4 Datos introducidos

3. Y por último, con los comandos `ruby -v` y `rails -v`, realizamos la comprobación de versiones instaladas en el sistema.

Los usuarios de la marca de la manzana mordida de Adán, los admiradores de la creatividad de *Steve Jobs* también pueden disfrutar de Ruby, y el apartado siguiente es para ellos. En este momento cada uno sigue su senda. Los usuarios de Windows, los seguidores de la universalización, la democratización del acceso a las aplicaciones tecnológicas pueden dirigirse a la página 13, donde comenzaremos con los Editores e IDE'S.

Instalación en Mac OS X

Hay que mencionar que en los sistemas Mac OS X viene preinstalado Ruby. Pero se trata de una versión desactualizada como es la 1.8.7. Se puede comprobar abriendo un terminal e introduciendo el comando `ruby -v`.

Pasemos a detallar el procedimiento para la instalación de RoR:

1. Es necesario tener instalada una herramienta que nos permita transferir archivos mediante comandos desde el terminal. Para ello la opción utilizada es `cURL`.
 - a. Con el comando `curl --version` desde un terminal, es posible confirmar la preexistente instalación de este. En tal caso, ya podemos continuar en el segundo punto.
 - b. Instalamos `cURL` desde la página oficial <http://curl.haxx.se/>.

The screenshot shows the official cURL website. The main content area is titled "Releases and Downloads". Under "Source Archives", it lists several curl versions with their release dates and download links for different operating systems (macos, windows, linux, solaris, freebsd, netbsd, openbsd, android, ios, vxworks, haiku, and darwin). A "Download Wizard" section offers help in selecting the correct download link. Below that, a "Packages" section provides a table of available packages for various operating systems.

Package	Version	Type	SSL	SSH	Provider	Size
AIX					AIX	
AIX 4.3 RPM	7.8.3	binary	lib			
AIX 4.3 RPM	7.8.3	binary		lib	Butt Enterprise	
AIX 4.3 RPM	7.8.3	source		lib	Butt Enterprise	
AIX 4.3 RPM	7.8.3	source	lib			
AIX 4.3.3 tar+Z	7.15.1	binary	SSL		The Written Word	1.71 MB
AIX 5.1 tar+Z	7.15.1	binary	SSL		The Written Word	1.69 MB

Fig. 1-5 Página oficial de cURL

2. **Xcode.** Es el entorno de desarrollo integrado (IDE) de Apple, que proporciona las herramientas necesarias para programar en nuestro Mac. Sumamente útil por el compilador de C que viene incluido. Dicho compilador lo necesitaremos para nuestros programas escritos en Ruby y que podemos descargar desde el *AppStore* y proceder a su instalación. Para ello debemos disponer de una cuenta de desarrolladores. En el caso de que se trate de la versión 4.2, apreciaremos problemas de integración con Ruby, por lo que nos veríamos obligados a realizar una actualización de la misma.

3. **Homebrew.** Es un gestor de paquetes para OS X, que podemos instalarlo desde nuestro terminal. La comprobación de su instalación previa se realiza a través de la introducción del comando `brew -v`.
- a. Si debemos instalarlo, emplearemos la herramienta `cURL` previamente instalada en el paso número uno y el siguiente comando que podemos encontrar en la página oficial <http://mxcl.github.io/homebrew/index.es.html> comenzando la descarga e instalación.

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/mxcl/homebrew/go)"
```

4. **Ruby Version Manager (RVM).** Es una herramienta que permite gestionar y trabajar con múltiples versiones de Ruby en nuestro sistema. Por ejemplo, podemos tener una aplicación con la versión 1.8.7 y otra con la 1.9.3 sin que surjan conflictos entre ellas.
- a. En la guía oficial existe el siguiente comando para iniciar su instalación con la herramienta `cURL`.

```
$ \curl -#L https://get.rvm.io | bash -s stable
```

```
// Cargar la configuración después de su instalación
```

```
$ rvm reload
```

- b. Dentro del `.bash_profile` debemos introducir una nueva línea para evitar conflictos dentro del terminal con las diferentes versiones. El primer comando indicado sirve para abrir el archivo en modo de edición, la segunda línea es la que deberemos insertar y así recargar con `source` los valores del archivo para que se hagan efectivos los cambios.

```
nano .bash_profile
```

```
[[-s"$HOME/.rvm/scripts/rvm" ]] && . "$HOME/.rvm/scripts/rvm"
```

```
source .bash_profile
```

- c. Confirmamos que todo se ha realizado correctamente con el comando `type rvm | head -1` en nuestro terminal. La respuesta *“rvm is a function”* es la confirmación de que así ha sido.
- d. Es entonces cuando podemos instalar la versión 1.9.3 de Ruby con `rvm`.

```
rvm install 1.9.3 // Instalar Ruby versión 1.9.3.
```

```
rvm list // Listar todas las versiones instaladas.
```

```
rvm --default use <versión> // Indicar qué versión utilizamos por defecto.
```

5. Por último nos quedaría instalar la gema Rails desde el terminal con el comando `gem install rails 3.2`.

Podemos dar por concluida la instalación de Ruby on Rails. Apréstense pues a disfrutar del éxito de su trabajo y a abrirse a nuevas, vivificantes y prácticas experiencias. No obstante y como guinda de nuestro pastel, podemos instalar, con la utilidad *Homebrew*, un sistema de control de versiones de nuestra aplicación con *git* (ver punto 1.1 donde explicamos su utilidad, con la descripción del pack de instalación de *RailsInstaller*).

```
brew install git //instalación de paquete git
```

Acuérdate también de crear una nueva cuenta de usuario de github desde su página oficial <http://www.github.com> (ver punto 1.1 paso segundo para más detalle).

NOTA

Para los usuarios que no les agrada la ejecución de comandos, tenemos disponible un pack que instala todo lo necesario, válido para sistemas operativos Mac OSX y Windows. <http://www.railsinstaller.org>.

Instalación en Linux

El abanderado del software libre tampoco se puede resistir a los encantos del framework Ruby on Rails, por ello iremos detallando paso a paso cómo integrarlo con nuestro Linux Ubuntu versión 12.

Debemos de tener instalado un gestor de paquetes, y centrarnos en el conocido apt-get.

1. **Repositorios.** En primer lugar actualizamos los repositorios de Ubuntu con el siguiente comando:

```
sudo apt-get update
```

2. **cURL.** Es necesario tener instalada una herramienta que nos permita transferir archivos mediante comandos. Para ello la opción seleccionada es *cURL*, incluimos también la instalación git (ver *Instalación en Windows donde explicamos su utilidad, con la descripción del pack de instalación de RailsInstaller*).

```
sudo apt-get install git curl
```

3. **Ruby Version Manager (RVM)**, es una herramienta que permite gestionar y trabajar con múltiples versiones de Ruby en nuestro sistema. Por ejemplo podemos tener una aplicación con la versión 1.8.7 y otra con la 1.9.3 sin que surjan conflictos entre ellas.

- a. En la guía oficial existe el siguiente comando para iniciar su instalación con la herramienta cURL.

```
$ \curl -#L https://get.rvm.io | bash -s stable
```

```
//Cargar RVM
```

```
source ~/.rvm/scripts/rvm
```

- b. **Dependencias.** Instalamos las dependencias necesarias que requiere RVM (Ruby Versión Manager).

```
rvm requirements
```

```
sudo apt-get -y install build-essential openssl libreadline6  
libreadline6-dev zlib1g zlib1g-dev libssl-dev libyaml-dev libsqlite3-0  
libsqlite3-dev sqlite3 libxml2-dev libxslt-dev autoconf libc6-dev  
ncurses-dev automake libtool bison subversion
```

- c. **Framework CSS.** Como es necesario para las nuevas versiones de Rails una *framework javascript* instalamos el llamado nodejs.

```
sudo apt-get install nodejs
```

- d. **Ruby.** Llegamos al punto donde estamos listos para instalar Ruby, y nos ayudamos de RVM.

```
rvm install 1.9.3 // Instalar Ruby versión 1.9.3.
```

```
rvm list // Listar todas las versiones instaladas.
```

```
rvm --default use <versión> // Indicar qué versión utilizamos por defecto.
```

Una vez finalizado con el comando `rvm -v` podemos verificar la versión instalada de Ruby.

4. **Instalar Rails.** Como último paso podemos instalar el framework como una gema.

```
gem install rails
```

Editores o IDE'S

Hay un punto en el que están de acuerdo físicos y filósofos, y es que la realidad es una construcción que todos y cada uno de los seres hace en su intelecto, y por tanto no hay ninguna verdad unívoca y absoluta (sí, efectivamente la frase en sí misma encierra su contradicción). Viene a colación porque como para todo en la vida, cada uno tiene su opinión, su gusto y su inclinación. Por ello, no es de extrañar, que cada uno tenga, también, su preferencia por un editor de código o IDE (Integrated Development Environment).

La diferencia que hay entre ellos es que con los IDE tendremos un compilador, un depurador y un constructor de interfaz gráfica (GUI), además de ser también un editor de código.

El autor, ejerciendo su legítima opción a la subjetividad, se ha inclinado por el editor Sublime Text 2, explicitando su favorable opinión del mismo sobre otros. No obstante, apreciando el más que respetable gusto del amable lector se exponen a continuación algunos de los más conocidos editores compatibles que se pueden probar para trabajar con Ruby on Rails.

LISTADO DE EDITORES E IDE'S

- Editor Textmate (Mac).
- Editor E-texteditor (Windows).
- Editor Gedit (Linux).

- IDE RadRails (Windows, Mac y Linux).
- Editor Vim (Linux y Mac).
- IDE Aptana RadRails (Windows, Mac y Linux).
- IDE RoRED (Windows).
- IDE NetBeans 7.0 (Windows).
- Editor Notepad ++. (Windows).
- IDE Komodo (Mac y Windows).

Es altamente recomendable sentirse familiarizado con el editor o IDE elegido. Permítanos recomendar el dedicar tiempo a leer la documentación si finalmente buscamos una alternativa a Sublime Text 2. Además, existen plugins que nos ayudarán a agilizar aún más la escritura de código. Siendo preciso para ello, realizar una labor de investigación al objeto de encontrar los más recomendables para cada uno de ellos. Teniendo siempre muy en cuenta lo innecesario de instalar todos los que encontremos y nos recomienden, y valorando los que verdaderamente nos sean útiles y necesarios. En el supuesto de que se siga la humilde recomendación del autor y se emplee SBT2, está disponible un vídeo de formación con las características básicas y las más avanzadas (consultar Introducción).

EDITOR SUBLIME TEXT 2

Está disponible una versión sin ninguna restricción sobre su uso, aunque no es gratuita y aleatoriamente conforme vamos grabando los cambios nos cuestiona sobre el interés en la adquisición de la licencia (coste 29\$). El autor, sin otro interés que el de el mero servicio, recomienda probarlo y adquirirlo en función de la satisfacción del usuario, permitiéndose destacar las siguientes ventajas.

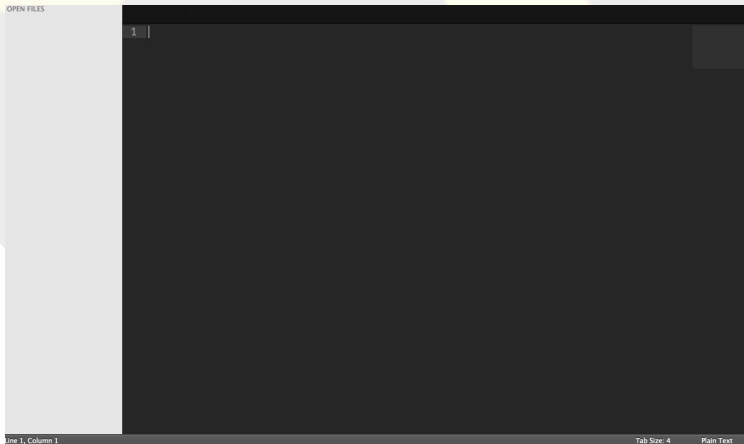


Fig. 1-6 Sublime Text 2

- **Comandos.** Consola intérprete del lenguaje de programación Python para desde ella realizar diferentes tareas.
- **Programación multiplataforma.** Soporta de forma nativa infinidad de lenguajes de programación y texto plano.
- **Interfaz.** Limpia, sencilla e intuitiva, con mini-mapas sobre el código, multi-layout, multi-pestañas y coloreado de código.
- **Búsquedas instantáneas.** Podemos ayudarnos de las expresiones regulares para las búsquedas de archivos y líneas concretas. Son realmente muy rápidas.
- **Personalización.** Es totalmente personalizable desde la configuración de las combinaciones de teclas, interfaz, colores, fuentes, tamaños, etc.
- **Snippets.** Creación de Snippets para el ahorro considerable de tiempo en nuestras implementaciones.

INSTALACIÓN Y CARACTERÍSTICAS

Para su descarga, accedemos a la página oficial <http://www.sublimetext.com>, donde primeramente debemos instalar un gestor de paquetes llamado *Package Control* que se integra perfectamente con nuestro editor como una utilidad más. Para ello acudimos a su web oficial http://wbond.net/sublime_packages/package_control/installation con la intención de copiar el comando en el apartado e insertarlo en la consola Python de Sublime Text2 (View -> Show Console).

```
import urllib2,os; pf='Package Control.sublime-package';
ipp=sublime.installed_packages_path(); os.makedirs(ipp) if not os.path.exists(ipp) else
None; urllib2.install_opener(urllib2.build_opener(urllib2.ProxyHandler()));
open(os.path.join(ipp,pf),'wb').write(urllib2.urlopen('http://sublime.wbond.net/'+pf.re
place(' ','%20')).read()); print('Please restart Sublime Text to finish installation')
```

Disponible para continuar con la explicación de Sublime Text 2, un vídeo en el que profundizaremos aún más (consultar Introducción).