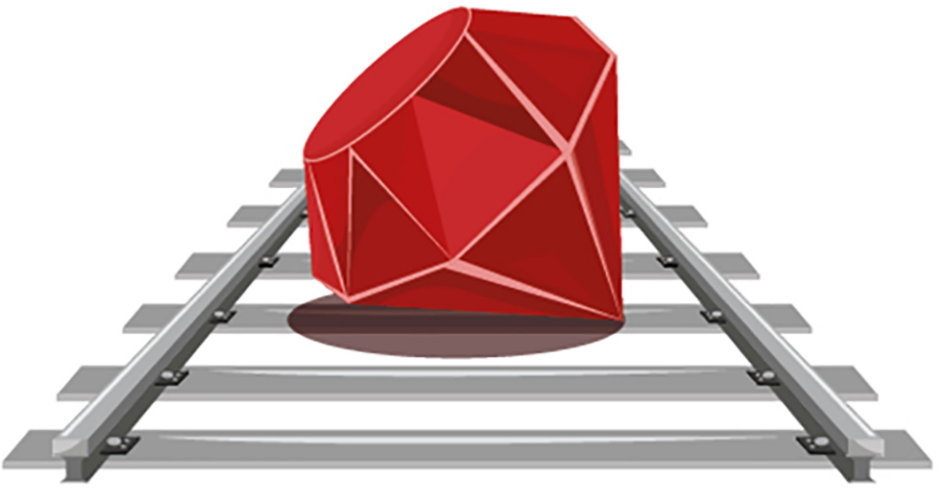
RUBY ON RAILS

Aprende a crear aplicaciones web desde cero



Javier Vázquez y Daniel Lorenzo

libros RJ

alphaeditorial

RUBY ON RAILS

Aprende a crear aplicaciones web desde cero

RUBY ON RAILS

Aprende a crear aplicaciones web desde cero

Javier A. Vázquez Olivares
Daniel Lorenzo Martínez

ALPHA EDITORIAL
ALFAOMEGA COLOMBIANA S.A.
Calle 62 No.20-46 esquina, Bogotá
Teléfono (57-1) 746 0102 Fax: (57-1) 210 0122
cliente@alfaomegacolombiana.com
www.alfaomega.com.co
www.alpha-editorial.com

RC LIBROS
Calle Mar Mediterráneo, 2. N-6
28830 San Fernando de Henares, Madrid
Teléfono +34 91 677 57 22
info@rclibros.es
www.rclibros.es

Primera edición: Madrid, 2020
Bogotá, 2020

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

ISBN: 978-958-778-636-1 (Colombia)
ISBN: 978-84-948972-9-0 (España)

Hecho en Colombia
Printed and made in Colombia

ÍNDICE

INTRODUCCIÓN

Sin duda alguna, Ruby y su framework Rails han despertado el interés de muchísimas personas alrededor del mundo. La razón que con más probabilidad ha causado esto es la necesidad de herramientas de desarrollo ágil que permitan al programador construir y depurar productos de software rápida y ordenadamente. En este aspecto, pocos frameworks de desarrollo web se comparan con Ruby on Rails, sin embargo, Rails no sería lo que es si no fuese por el lenguaje de programación que lo soporta. Cuando combinas la versatilidad y rapidez de Rails con la elegancia y legibilidad del lenguaje Ruby tienes como resultado programadores felices y productivos.

Cuando llegues a dominar Ruby on Rails, es posible que puedas realizar en la mitad del tiempo, lo que otro programador haría en otra tecnología. Ruby on Rails es simplemente genial.

En este libro aprenderás paso a paso cómo crear software con Ruby on Rails desde cero. Comenzaremos con las bases del lenguaje Ruby y paulatinamente irás aprendiendo temas más avanzados. Te darás cuenta de que Ruby es intuitivo e idiomático, esto quiere decir que programar en Ruby se parece mucho al lenguaje natural (en inglés) y la estructura de código es tan sencilla que por momentos parece que escribimos en pseudocódigo.

Con explicaciones sencillas y varios ejemplos que podrás practicar por ti mismo, aprenderás sobre variables, objetos, arreglos, hashes y otros temas de importancia en Ruby.

El libro se compone de dos secciones. En la primera sección verás los temas relativos al lenguaje Ruby. En la segunda sección, aprenderás sobre su framework Rails. En esta segunda sección, aprenderás las bases del desarrollo web, servidores web, protocolo HTTP, la filosofía y formas de trabajo en Ruby on Rails, sus ventajas respecto a otras plataformas y de inmediato comenzaremos a desarrollar nuestras primeras aplicaciones paso a paso e incrementando el nivel paulatinamente conforme avanzamos. Y como comprador tienes acceso a los anexos complementarios de este libro que pueden ser descargados gratuitamente, visitando: http://rclibros.es, y después su página individual.

Al terminar este libro/curso, serás capaz de realizar potentes aplicaciones web, construir servicios API Rest y montar tus servicios o aplicaciones en un servidor en la nube.

LOS AUTORES

Javier Arturo Vázquez está comprometido con el mundo del desarrollo de sistemas desde hace 8 años, ha participado como ingeniero de software para las fuerzas armadas de México y algunas empresas de ámbito privado, desarrollando simuladores, sistemas cliente-servidor y sistemas web, utilizando diversas tecnologías y lenguajes de programación. Es un apasionado de la enseñanza y cuenta con varios cursos en la plataforma udemy.com, además de un canal en youtube (profe Javier) donde enseña programación en Java y Ruby principalmente. Actualmente colabora en una empresa emergente como director de desarrollo utilizando Ruby on Rails y varias otras tecnologías para la construcción de una plataforma web y móvil de impacto mundial.

Daniel Lorenzo Martínez es ingeniero y licenciado en sistemas por el Instituto Tecnológico de Misantla de Veracruz, México; a lo largo de su carrera, ha trabajado en el desarrollo de productos de software con varios lenguajes y tecnologías, especialmente Ruby y Rails; su actividad está relacionada con el desarrollo de sistemas de facturación electrónica y web con fines educativos, actualmente trabaja como desarrollador de proyectos para una importante empresa americana utilizando técnicas de repetición espaciada para la memorización.

Parte I

RUBY

MI PRIMER ENCUENTRO

Esta parte se compone de cinco capítulos. En los primeros tres conocerás las características de este maravilloso lenguaje, lo instalarás y harás tus primeros programas. Luego habrá un capítulo en donde explicaremos de manera conceptual y con ejemplos del mundo real en qué consiste la programación orientada a objetos (POO), ya que Ruby se caracteriza por ser 100% orientado a objetos. Y el último capítulo será prácticamente un paseo por la API de Ruby.

1

ACERCA DE RUBY

Permíteme hablarte un poco sobre el lenguaje y desmitificar también algunos aspectos en este capítulo. Trataré de ser objetivo para darte una descripción lo más certera posible para que con ello estés plenamente consciente de lo que estás próximo a aprender.

SOBRE EL LENGUAJE Y SU CREADOR

A este precioso y valioso lenguaje (como la piedra que lo representa) lo que lo hace precioso y de gran estima es su naturalidad. Yukihiro Matsumoto ha logrado crear un lenguaje con una envidiable facilidad de lectura, tanto que desde que hagas tu primer "hola mundo" verás su gran naturalidad.

Para entender la razón que llevó a Matz a hacer su propio lenguaje, debes de tener en cuenta que él era experimentado en los lenguajes: Perl, Smalltalk, Eiffel, Ada y Lisp. Cuando Matz usaba uno en particular no se sentía satisfecho, como seguramente nosotros tampoco nos hemos sentido complacidos en su totalidad al usar alguno. Él admiraba la belleza y el potencial de cada uno, pero siempre había algo que lo desilusionaba y, cuando usaba otro, ocurría lo mismo. La gran mayoría de las veces, su insatisfacción fue que los lenguajes no eran muy expresivos, su lectura era posible únicamente por gente experimentada en el lenguaje. Este fue el mayor descontento y el motivo suficiente que lo llevó a crear el lenguaje Ruby. Matz aprovechó lo mejor de cada uno. Mientras creaba Ruby, tú y yo estuvimos en su mente, en el corazón de él estaba darnos a conocer un lenguaje, ¡sí!, pero más que un lenguaje, en realidad quería presentarnos a un amigo. Matz quiere que Ruby sea tu amigo y yo me encargaré de hablarte sobre él.

Ruby ya tiene sus ayeres pues nació en la década de los noventa, para ser exactos, en 1995 (fecha en la que se publicó) pero su creación llevó dos años. Una vez que lo concretó y lo publicó, enseguida se hizo popular.

Desde los primeros años, Ruby fue todo un éxito, fue muy bien recibido y además lanzado en el momento oportuno (en los inicios de la Web), justo por eso es que existen numerosos *frameworks* y *microframeworks* para este lenguaje. Por supuesto, el más popular de ellos es Ruby on Rails (el *framework* de aplicaciones web preferido por los *rubyist*). Ruby On Rails salió diez años después que Ruby, pero marcó la historia al llevar el florecimiento de Ruby a gran escala. De esta forma, Ruby fue abriéndose brecha entre los lenguajes más posicionados y atrayendo a más adeptos con sus *frameworks* como si fuesen invitaciones coloridas.

CARACTERÍSTICAS Y COMPARACIÓN

Aunque hay muchas tecnologías y lenguajes que nos permiten el desarrollo web, comparemos Ruby con dos de los lenguajes más populares para el desarrollo web: Java y PHP. De antemano aclaro que no tengo la intención de hablar mal de un lenguaje solo para engrandecer a otro, por favor ten esto muy presente.

LENGUAJES DE PROGRAMACIÓN

CARACTERÍSTICA	 RUBY	 JAVA	 PHP
Interpretado	☺	☹	☺
Multiparadigma	☺	☺	☺
Libre y de código abierto	☺	☹	☺
Rápido (velocidad de procesamiento)	☹	☺	☹
Legible	☺	☹	☹
Popular	☺	☺	☺
Multiplataforma	☺	☺	☺
Fácil de aprender	☺	☹	☺
Con buena documentación	☺	☺	☺
Con una comunidad activa y creciente	☺	☺	☺
Propósito general	☺	☺	☺

Hay algunas características que ya las mencioné anteriormente de forma indirecta, pero hay otras que necesitan ser definidas. Por ejemplo:

Interpretado: Cuando escribimos código lo hacemos en un lenguaje muy similar al nuestro –en este caso usando Ruby– pero la computadora no entiende ni una *jota* de lo que queremos que haga. Para darle a conocer a nuestra computadora las instrucciones en su lenguaje, necesitamos de un procesador de lenguaje que actúe como un intermediario entre nosotros y nuestra computadora. Los procesadores de lenguaje son programas que convierten el código fuente en lenguaje máquina (ceros y unos). Los procesadores de lenguajes pueden ser compiladores o intérpretes.

En un lenguaje interpretado el código fuente será convertido al lenguaje máquina en el momento de ejecución por el intérprete, mientras que en un lenguaje compilado requerirá convertir el código fuente en su totalidad a lenguaje máquina, antes de ser ejecutado.

Ruby es un lenguaje interpretado y "el intermediario entre nosotros y Ruby" se llama *Interactive Ruby Shell (irb)*.

Rápido (velocidad de procesamiento): El que Ruby use un intérprete no son buenas noticias en cuanto al tiempo de ejecución, ni para él ni para algún otro lenguaje que funcione de esta manera.

En conclusión, con un lenguaje interpretado estamos sacrificando velocidad por comodidad, pero descuida: Ruby tampoco es lento como un caracol, si así fuera nadie lo querría usar.

Multiparadigma: Ruby es multiparadigma. Los *paradigmas de programación* son propuestas aceptadas como modelos que pretenden resolver los problemas en el desarrollo de software bajo la influencia de ciertas filosofías. Ruby acepta la programación imperativa y la funcional, pero, aunque es posible programar de forma funcional, lo suyo son los objetos. La *programación orientada a objetos* (POO) es sin duda una de las características principales de Ruby y consiste en imitar a los objetos del mundo real y hacer de ellos objetos de programación, objetos que al igual que los del mundo real tienen características y comportamientos.

No te preocupes ahora por entender este paradigma. Un poco más adelante te explicaré en qué consiste, utilizando analogías del mundo real.

Libre y de código abierto: Ruby puede ser usado por cualquiera sin ningún costo. Y del código fuente del lenguaje puedes hacer lo que te plazca, si quieres puedes adaptarlo a tus necesidades, resolver bugs, realizar alguna mejora, distribuirlo, etc. Tienes toda la libertad concedida.

Multiplataforma: Como consecuencia de ser un lenguaje interpretado, tus programas en Ruby podrán ser ejecutados en cualquier sistema operativo (en

Windows, en alguna distribución GNU/Linux o Mac Os) siempre y cuando tengas instalado el intérprete de Ruby.

Fácil de aprender: No es que está característica necesite de una definición clara y detallada, sino que veo una oportunidad para hacer una aclaración sobre los comentarios precipitados y endebles que se han suscitado por la "simplicidad" del lenguaje y que han influido en muchas personas a tal grado de rechazarlo por la falsa imagen que se ha creado en algunos círculos de programadores. Ruby no es un lenguaje para novatos o peor aún como lo expresan muchos: *"un lenguaje para niñas"* (lo que evidencia el poco criterio de los que lo comentan pues las niñas también pueden ser excelentes programadoras), no, Ruby es un lenguaje muy maduro enfocado en la productividad. Es cierto que se recomienda aprender a programar con lenguajes como PHP, Python o Ruby para luego pasar a lenguajes más "serios" como C++ o Java pues con estos lenguajes de programación, prácticamente el mismo día que lo instalas puedes estar escribiendo código muy fácilmente. Esto nos habla de lo mucho que podemos lograr en muy poco tiempo, no de una simpleza.

Con buena documentación y una comunidad activa: Uno de los temores al iniciarse en algún lenguaje, es avanzar y llegar a un punto sin salida. En más de una ocasión nos ha pasado esto ¿verdad? ¡No hay nada que temer! La API de Ruby es una de las mejores documentadas y cuenta con una gran cantidad de ejemplos.

Puedes echar un vistazo a la documentación oficial de Ruby en la siguiente dirección: https://ruby-doc.org/

Generalmente la documentación oficial resolverá un montón de dudas, pero, si aún hubiese algo que no entendemos, en *ruby-talk* (es la lista de correos más popular donde se tratan temas en general de Ruby) siempre habrá alguien dispuesto a responder a tus preguntas. Para poder suscribirte a la lista de correo *ruby-talk* escribe la siguiente url en tu navegador.

https://www.ruby-lang.org/es/community/mailing-lists/

De propósito general: Aunque su framework Rails se ha encargado de que sea mayormente utilizado para aplicaciones web, Ruby puede tener diferentes usos y aplicaciones. Si quieres saber al respecto, te dejo un enlace que te llevará a una lista de historias de éxito. Algunas de estas historias tienen que ver con simuladores, modelado 3D, robótica, redes, entre otras más. Solo por ponerte un ejemplo, el centro de Investigación Langley de la NASA y Motorola usaron Ruby para crear simuladores. Además, el recientemente popular google Sketchup utiliza Ruby como motor para el diseño y modelado 3D.

https://www.ruby-lang.org/en/documentation/success-stories/

2

PREPARACIÓN DEL ENTORNO DE TRABAJO

Las instalaciones en general son una de las cosas que muchos de nosotros odiamos, pero sea que queramos o no, las debemos hacer ☹. Sin embargo, si sigues los pasos que describiré a continuación según el sistema operativo de tu preferencia, en máximo 20 minutos tu computadora estará ansiosa por trabajar 😋.

Posterior a la instalación de Ruby y Rails, tendrás tiempo para ir probando con libertad las herramientas de edición de código que te mencionaré u otra que desees conocer. Al final usa la que más te agrade o en la que mejor te desempeñes.

INSTALACIÓN DE RUBY Y RUBY ON RAILS

Los ejemplos que se incluyen en el presente libro en su totalidad fueron escritos usando **Ruby 2.4.4** (la versión estable y con mayor cantidad de gemas compatibles poco antes de la publicación de este libro) y **Rails 5.2.2**. Nosotros estamos usando estas dos versiones, pero no debe haber cambios significativos en una versión mayor. Aun así, mi deseo es que me sigas a la par sin ningún problema hasta el final del libro y para eso, te mostraré cómo lo hice yo.

En la siguiente dirección podrás elegir cualquier versión de Rails:

https://rubygems.org/gems/rails/versions

Una cosa es cierta, no todos contamos con los mismos equipos, ni sistemas operativos (de ahora en adelante SO) en nuestras PCs, por esa razón mostraré la instalación del lenguaje, de su framework y de todo lo necesario, tratando de cubrir

la mayor cantidad de SOs. No es necesario que leas todo, ve directamente a la explicación de la instalación del SO que uses, ya sea Windows, alguna distribución de GNU/Linux basada en Ubuntu u OS X.

Instalación en Windows

A pesar de que existe un empaquetado para Windows que nos haría la vida de maravilla llamado *RailsInstaller*, en este apartado te enseñaré a instalar Ruby y Rails en las versiones propuestas (2.4.4 y 5.2.2, respectivamente). La ventaja de hacerlo de esta manera sin duda alguna es que podrás instalar la versión que consideres más conveniente.

A pesar de lo dicho, si quieres hacerlo por medio de *RailsInstaller*, el ejecutable lo encontrarás en la página oficial, *RailsInstaller*.

http://railsinstaller.org/en

La instalación es sumamente sencilla, prácticamente es siguiente, siguiente, siguiente y ¡ya! (aunque por lo general todas las instalaciones así son).

Para hacerlo de la segunda forma primeramente debes dirigirte a la página oficial de *RubyInstaller* y dar clic en "Download" (un botón que resalta a primera vista). Te redireccionará a una nueva página donde se encuentran los instaladores de Ruby en sus diversas versiones.

El enlace a la página para realizar la descarga es el siguiente:

https://rubyinstaller.org/

En la lista de versiones disponibles, encontrarás dos grupos: *Ruby sin Devkit* y *Ruby con Devkit*, deberás descargar: *Ruby + Devkit 2.4.4* con terminación en *x64* si tu computadora es de 64 bits o con terminación *x86* si tu computadora es de 32 bits. Si no estás seguro, puedes ir al panel de control y ver la información de tu SO.

Toma en cuenta que la página se actualiza constantemente ya que salen nuevas versiones. Por lo tanto, en la página del enlace anterior solo mostrará las versiones más recientes, pero sería bueno que leyeras lo que en ella se encuentra, te será de mucha ayuda. Si deseas ver alguna versión más antigua deberás hacer clic en el enlace "Archives" y te direccionará a un listado más amplio donde encontrarás desde la versión 1.8.7 hasta la más actual y estable.

La página de RubyInstaller es la que mostraré a continuación. En ella está señalada con una flecha el enlace mencionado.

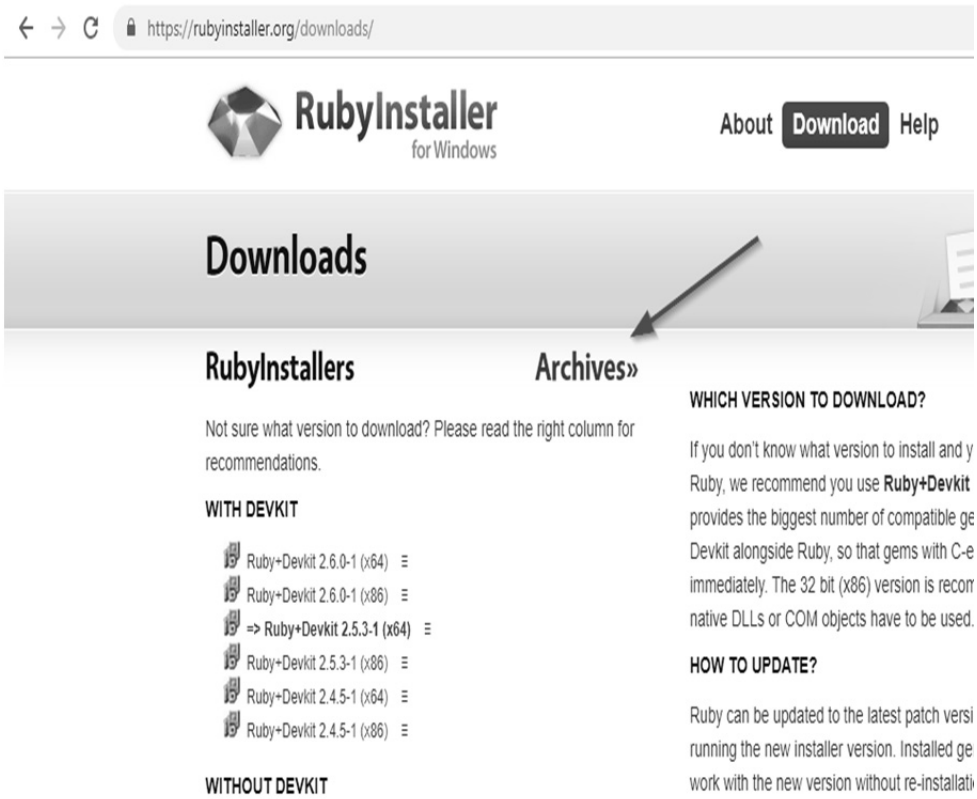


Fig. 1 Página oficial para la descarga del instalador de Ruby para Windows

Cuando se haya descargado el instalador de la página web por completo, deberás de ejecutar el instalador haciendo doble clic sobre él, esto hará que muestre el asistente de instalación. Cuando aparezca, deberás de aceptar los términos de licencia y dar clic en “Next”.

Deja que el asistente instale Ruby en la ruta que se muestra, asegurándote de que las casillas de verificación “Add Ruby executable to your PATH”, “Associate.rb and .rbw files with this Ruby instalation” y “Use UTF-8 as default external encoding” estén marcadas para continuar con la instalación.

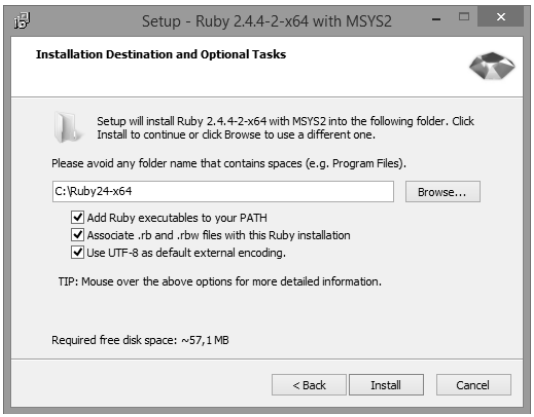


Fig. 2 Destino de instalación de RubyInstaller

Debido a que descargamos Ruby con Devkit debes de dejar seleccionada la casilla "MSYS2 development toolchain 2018-06-24" y dar clic en "Next".

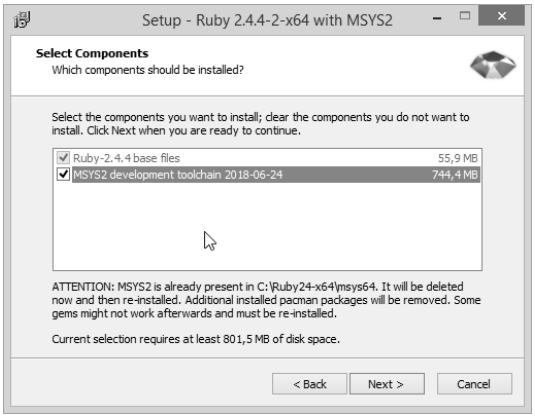


Fig. 3 Instalación del componente MSYS2

Después, habrá una barra de progreso para informarnos el tiempo faltante de forma gráfica. Espera a que esta termine. La instalación concluida merece el par de minutos de espera.

Hemos instalado Ruby en Windows, eso nos podría bastar, sin embargo, puedes comprobarlo con el comando *ruby -v* en el símbolo del sistema o CMD.

Para la instalación de Rails simplemente debes de escribir en el símbolo del sistema el comando *gem install Rails* y esperar pacientemente hasta que finalice la instalación. Sin embargo, te podrás dar cuenta en la imagen que sigue después de esta que este comando instala la versión de Rails más reciente y estable.

Si lo haces de esta manera, después de ingresar el comando anterior tu terminal se verá similar o igual a lo siguiente:

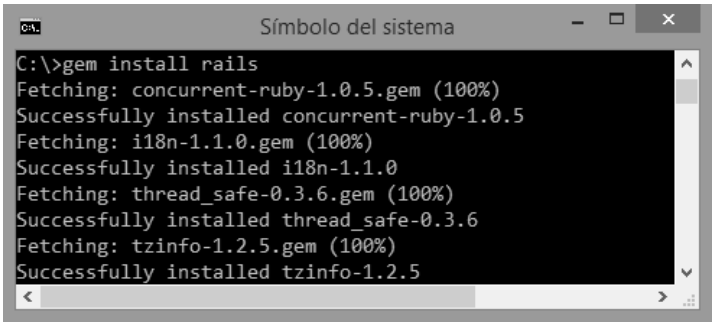


Fig. 4 Instalación de Ruby on Rails (Windows)

Durante la instalación hay un momento en el que pareciera que no avanza, pero no interrumpas la instalación, te comento que esto puede retrasarse un par de minutos, sé paciente. 😊

Al igual que para Ruby, puedes comprobar la versión de Rails que se instaló con el comando *rails -v*.

Si quieres instalar una versión en concreto, por ejemplo, la versión 5.2.2, entonces escribe el comando ***gem install rails -v 5.2.2***. Donde 5.2.2 es la versión a instalar, sustitúyela por la que desees. Ya sea que instales precisamente esta o la más reciente y estable en teoría no deberías de tener ningún problema, pero si quieres sentirte seguro, nosotros usaremos la **versión 5.2.2.**

Instalación en GNU/Linux

Linux es un SO muy amado por los desarrolladores y probablemente ya te lleves muy bien con la *terminal*, por lo tanto, escribir comandos no debería hacerte entrar en pánico. Si tienes Windows y a pesar de ello quieres mejor trabajar en Linux, te mostraré cómo instalar todo lo necesario para poder trabajar con Ruby on Rails en *Elementary OSX 0.4.1 Loki*. Este SO está basado en *Ubuntu 16.04.2* y es una buena opción para introducirse en el mundillo de Linux. Además, le va a caer muy bien a tus ojos ya que tiene un buen diseño.

Pero lo anterior no quiere decir que la siguiente guía de instalación esté dedicada sola para los que tengan *Elementary* OS. Si tu distribución favorita está basada en Ubuntu, podrás escribir los mismos comandos en tu terminal y tendrás con toda seguridad Ruby y Rails “al cien”.

Ahora bien, para la instalación de Ruby bastaría instalar algunas dependencias y escribir en la terminal un par de comandos y ¡listo!, pero antes de instalar Ruby instalaremos RVM (*Ruby Version Manager*) para poder administrar múltiples versiones de Ruby. El tiempo que ahora inviertas se lo agradecerás después a *RVM*. *RVM* te permitirá tener más de una versión de Ruby en tu computadora y usar la que tú le indiques en un momento dado. De esta manera podrías tener un proyecto basado en una versión "X" y otro en otra versión sin que estas interfieran.

Antes de instalar RVM deberás teclear el comando *sudo apt-get install software-properties-common* en la terminal, por cierto, la terminal de Elementary OS se llama *Pantheon* y te comento esto porque puede resultarte un poco extraña. Esto es necesario para poder agregar un repositorio PPA. PPA significa archivo de paquete personal y esos archivos están alojados en un repositorio de software específico.

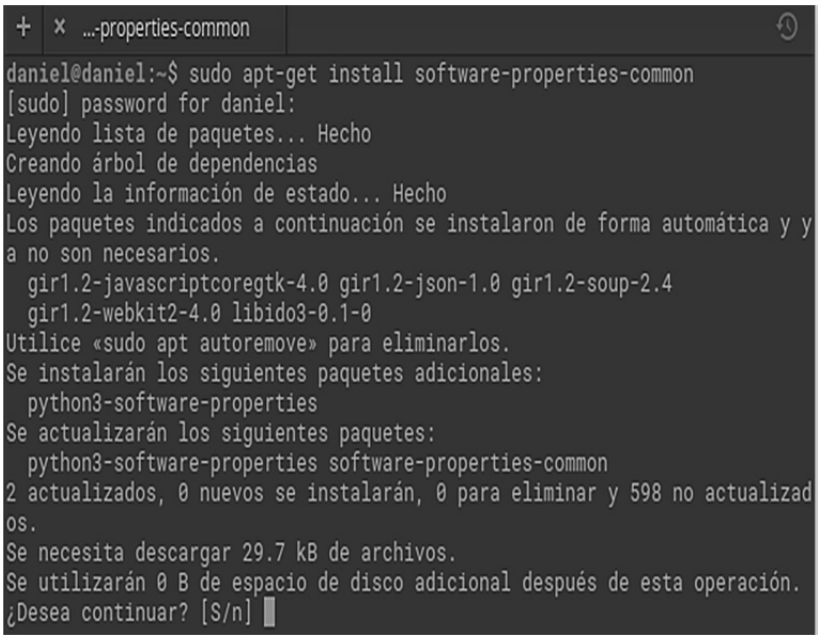


Fig. 5 Instalación de *software-properties-common*

Después de escribir el comando y de presionar enter, te preguntará si deseas continuar. Para confirmar que sí, deberás de ingresar una *S* y pulsar nuevamente la tecla *Intro*.

Para poder agregar el PPA de RVM deberás escribir *sudo apt-add-repository -y ppa:rael-gc/rvm.*

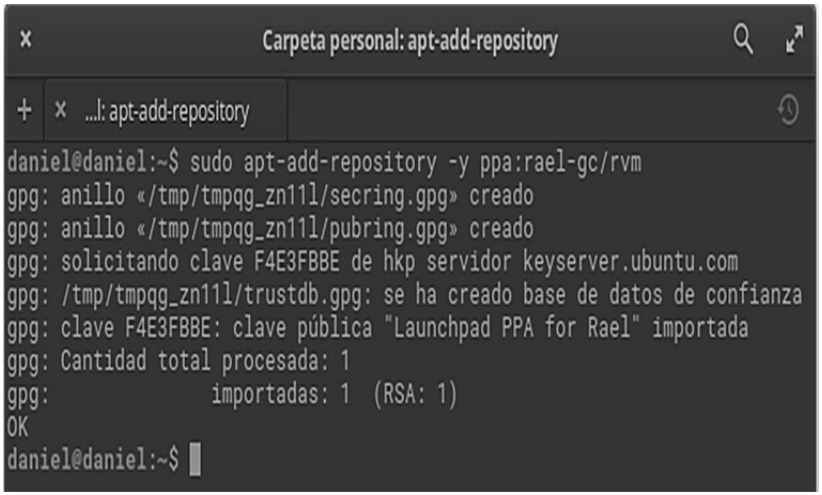


Fig. 6 Comando para agregar el PPA de RVM

Posteriormente deberás actualizar los repositorios con el comando bien conocido por todos, el comando *sudo apt-get update* como verás a continuación:

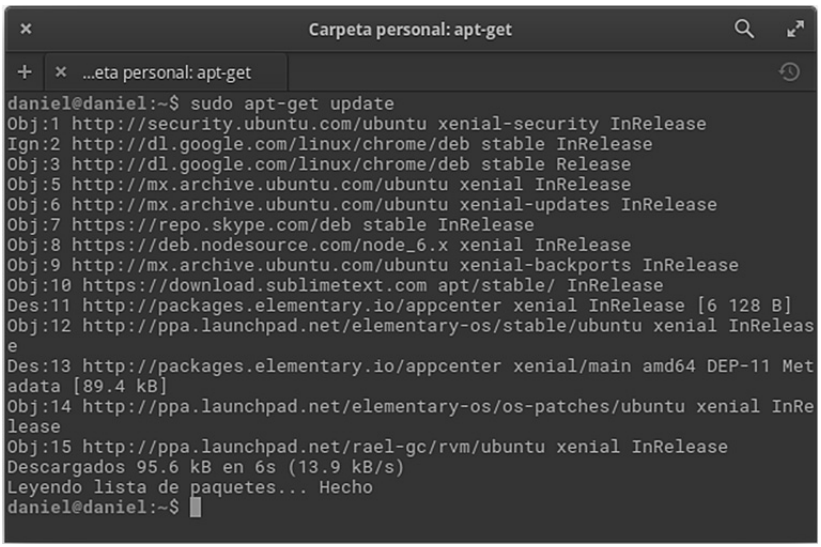


Fig. 7 Actualización de repositorios

Una vez actualizados los repositorios, procedemos a instalar *RVM* con *sudo apt-get install rvm.*

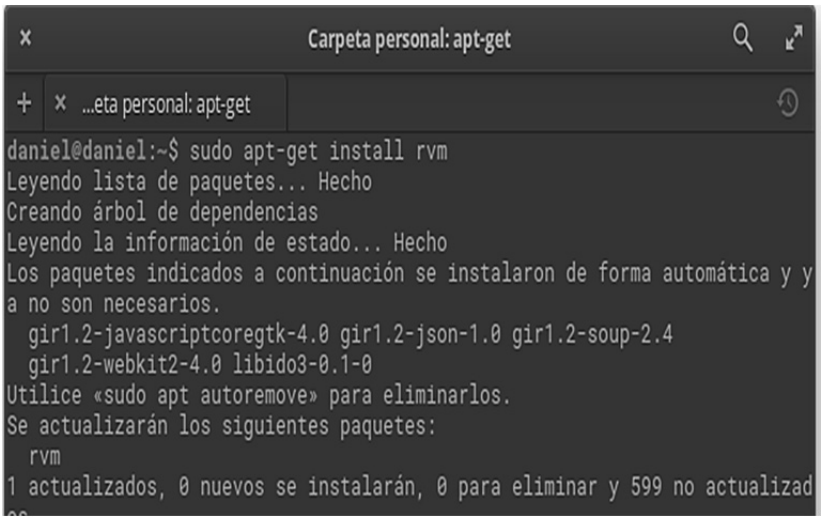


Fig. 8 Instalación de RVM

En este punto deberás de reiniciar tu computadora por los cambios efectuados para que todo funcione correctamente.

Abre nuevamente la terminal y escribe *source ~/.rvm/scripts/rvm* para cargar RVM.

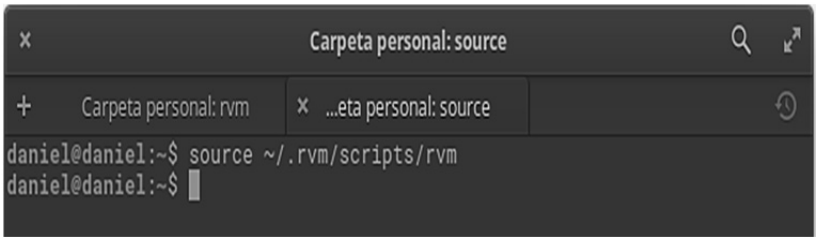


Fig. 9 Carga de RVM como función

Ahora ya podrás instalar las versiones de Ruby que gustes. Para la versión de *Ruby 2.4.4* escribe: *rvm install 2.4.4*. Mientras se instala puedes ir por un café y tomarte unos cinco o diez minutos, en ese tiempo seguramente ya habrá terminado.

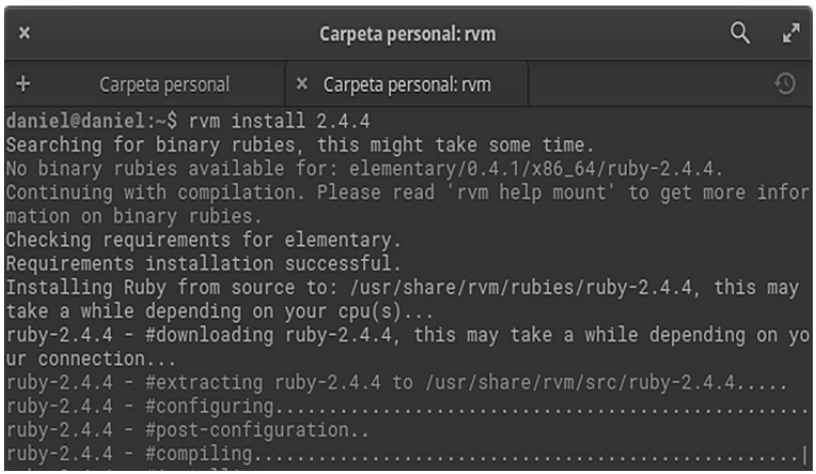


Fig. 10 Instalación de Ruby 2.4.4 con RVM

Si quieres ver las versiones de Ruby que se han instalado en el directorio de RVM, *rvm list* te mostrará todas ellas. Por ser tu primera vez solo te mostrará una, la que hayas instalado. Sin embargo, te mostraré las que yo tengo instaladas.

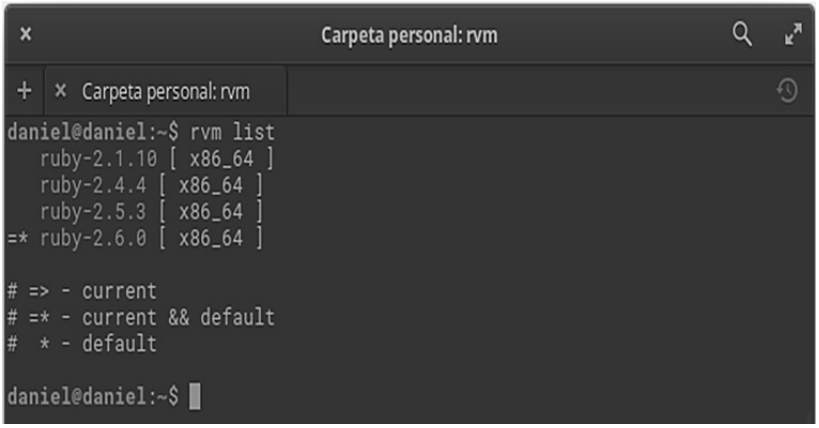


Fig. 11 Lista de las versiones de Ruby instaladas

Al ser la primera vez que se instala RVM y la única versión de Ruby en su lista, no hace falta indicarle que use una versión "X", pero en cuanto añadas una segunda versión (como yo), sí será necesario especificar la que quieres usar. Para hacer esto, escribe el comando *rvm use*, seguida de la versión que quieres usar. Haré esto con la versión 2.5.3.