Introducción a Ruby

¡Comencemos!

¿Qué aprenderás?

- Conocer la historia de Ruby.
- Conocer la "posición" de Ruby dentro de los lenguajes más conocidos.
- Conocer ventajas de Ruby sobre otros lenguajes.
- Conocer el ecosistema de empresas que utilizan Ruby.
- Conocer distintas versiones de Ruby y cuál utilizaremos en el curso.
- Conocer el rol de RVM.
- Instalar RVM en Ubuntu y macOS.
- Instalar Ruby en Windows.
- Instalar Ruby utilizando RVM en Ubuntu y macOS.
- Seleccionar una versión de Ruby utilizando RVM.

¡Vamos con todo!

Ruby

Ruby es un lenguaje de programación muy potente y flexible. Fue creado por Yukihiro Matsumoto y su primera versión fue lanzada en el año 1996. Desde entonces, ha sido adoptado por miles de empresas por su capacidad expresiva y cientos de herramientas que son fáciles de incorporar.

Ruby permite:

- Construir de forma sencilla aplicaciones web con manejo de bases de datos.
- Hacer análisis y visualización de datos.
- Hacer scraping (captura de datos de una página web).
- Crear juegos de video.
- Construir aplicaciones de escritorio.

Particularmente Ruby es famoso por su capacidad para construir aplicaciones web junto a Ruby on Rails.

¿Es Ruby una buena elección para comenzar a programar?

La respuesta es sí.

Ruby es un lenguaje que tiene muchas ventajas interesantes. Está pensado en la felicidad del programador, permite hacer muchas cosas con pocas líneas de código y su sintaxis es tan sencilla que permite ser leído de una forma muy similar al inglés.

Ruby en acción

Por ejemplo, ¿qué crees que hace la siguiente expresión?

```
3.times { print "hey hey hurray" }
```

O ¿Cuál crees que será el resultado de la siguiente expresión?

```
[1, 2, 3, 4].sum
```

Efectivamente, esta expresión entrega la suma de los números. Esto es lo que convierte a Ruby en una excelente opción para comenzar a programar.

¿Es Ruby relevante en la industria de desarrollo?

Ruby es la pieza clave detrás de Ruby on Rails, un framework para construir aplicaciones web. Hoy en día existen muchos lenguajes y frameworks para este propósito, las diferencias suelen ser bastante amplias en lo que respecta a sus propósitos y usos.

¿Qué empresas utilizan Ruby o Ruby on Rails?

Estas son algunas de las empresas más conocidas pertenecientes al ecosistema de empresas que utiliza Ruby on Rails como framework en su plataforma:

1. Airbnb
2. Github
3. SlideShare
4. Soundcloud
5. Couchsurfing
6. Groupon
7. Kickstarter
8. Twitch
9. Scribd
10. Shopify

Incluso la plataforma académica que ocuparemos para este curso fue desarrollada en Ruby on Rails.

¿Por qué aprender Ruby y no otro lenguaje de programación como Java, C# o JavaScript?

Todos los lenguajes de programación tienen sus ventajas, y con el tiempo y la experiencia estas diferencias comienzan a notarse más.

Ruby tiene varias ventajas sobre muchos otros lenguajes:

- Tiene una comunidad muy activa, unida y dispuesta a ayudar.
- El lenguaje está pensado para ser fácil de leer y de escribir.
- Existen miles de componentes fáciles de integrar, sistemas de autenticación, construcciones de gráficos, formularios, manejo de sprites para videojuegos, manejo de base de datos, etc.

Estas ventajas en conjunto permiten a equipos pequeños de desarrollo construir grandes proyectos.

Desventajas de Ruby

Ruby también tiene una desventaja, y es que no es un lenguaje particularmente eficiente. Por ejemplo, si quisiéramos hacer millones de cálculos matemáticos sería mejor trabajar con C.

Ruby y sus versiones

Tenemos que saber que Ruby ha tenido muchas versiones, y esto es importante porque las versiones no son compatibles entre ellas.

Breve historia de las versiones de ruby

Versión	Latest teeny version	Initial release date
1.0	NA	1996-12-25
1.8	1.8.7-p375[52]	2003-08-04
1.9	1.9.3-p551[56]	2007-12-25
2.0	2.0.0-p648[60]	2013-02-24
2.1	2.1.10[62]	2013-12-25
2.2	2.2.10[68]	2014-12-25
2.3	2.3.7[72]	2015-12-25
2.4	2.4.4[75]	2016-12-25
2.5	2.5.1[77]	2017-12-25
2.6		2018-12-25
3.0		2020

Tabla 1. Versiones Ruby.
Fuente: Desafío Latam

Versionamiento semántico

Ruby a partir de la versión 2.1 ocupa el versionamiento semántico. Esto quiere decir que:

- El primer número indica un cambio mayor, los proyectos no serán compatibles entre versiones mayores.
- El segundo número indica un cambio menor, actualizar esto puede requerir un esfuerzo menor, pero si el proyecto es muy grande de todas formas implica un gran esfuerzo.
- El último número indica un cambio pequeño, usualmente consiste en parches de seguridad que no romperán el funcionamiento de nuestro proyecto.

¿Qué versión utilizaremos?

En este módulo utilizaremos la versión 2.5.3 debido a que es la última versión estable disponible. Es importante, al momento de buscar documentación, utilizar versiones superiores a la 2.0 ya que el cambio es muy radical respecto a las anteriores y muy pocas cosas funcionarán.

No confundir Ruby con Ruby on Rails

Por otro lado, no tenemos que confundir Ruby, el lenguaje de programación, con Ruby on Rails, el framework para construir aplicaciones web. Cada proyecto tiene sus propias versiones. Rails va en la versión 5.2

El problema de una única versión

Porque no siempre estaremos trabajando en proyectos desde cero y, a veces, tendremos que trabajar sobre proyectos ya armados para agregar alguna funcionalidad o corregir un bug.

También veremos que muchos proyectos fijan una versión de Ruby para trabajar.

Ejemplo

Imaginemos que estamos trabajando en 3 proyectos de manera simultánea:

1. Proyecto de un cliente que se encuentra en Ruby 2.3 donde debemos agregar una funcionalidad nueva.

2. Proyecto que queremos revisar que se encuentra en Ruby 2.4.

3. Proyecto propio que estamos desarrollando en Ruby 2.5.

¿Cómo podemos manejar distintas versiones de Ruby?

Un gestor de versiones resuelve el problema. Este es un programa que permite tener instalado varias versiones de Ruby.

Existen varios gestores:

- RVM (Ruby Version Manager).
- Rbenv.
- Chruby.

Todos estos tienen el mismo propósito, pero su uso es ligeramente distinto. Nosotros trabajaremos con RVM.

Desventaja del gestor

No hay desventaja de ocupar un gestor, pero sí es un poco más difícil de instalar. Una posible desventaja es que no existe para el sistema operativo Windows.
Instalando Ruby en Windows

Instalando Ruby en Windows

Instalar Ruby en el sistema operativo Windows en bastante sencillo:

1. Ingresar a rubyinstaller.

2. Descargar el instalador.

3. Seguir las instrucciones.

Ruby funciona bien en Windows pero no todas las gemas de Ruby funcionan.

Instalando RVM y Ruby en Ubuntu y macOS

La instalación requiere de varios pasos. Las instrucciones actualizadas las podemos encontrar en instalar rails.

Seleccionando una versión de Ruby en Ubuntu y macOS

Una vez finalizada la instalación podemos listar la(s) version(es) de Ruby instalada(s) con la siguiente instrucción:

```
rvm list
```

Si la versión se encuentra listada podemos seleccionarla con rvm use versión.

```
rvm use 2.5.3
```

Si no se encuentra podemos instalarla con:

```
rvm install 2.5.3
```

Versión por defecto

Se puede dejar una versión de Ruby por defecto especificando --default.

```
rvm use 2.5.3 --default
```

Al seleccionarla deberíamos ver en la terminal lo siguiente:

```
Using ~/.rvm/gems/ruby-2.5.3
```