



## Fascículo 4

## Programación en diversos lenguajes

## Cuaderno 5: Programando libremente

## Ruby

Ruby es un lenguaje de programación creado por el programador japonés Yukihiro Matsumoto: comenzó a trabajar en él en 1993 y lo presentó públicamente en 1995 como "el mejor amigo del programador". Le puso el nombre de Ruby (en español rubí) como broma, aludiendo al lenguaje de programación Perl (perla).

Los principios que guiaron a su creador fueron la **productividad y la diversión para el desarrollador** y una **buena interfaz de usuario**, ya que consideraba que el diseño de programas necesita enfatizar las necesidades humanas más que las de la máquina.

Lo cierto es que Ruby combina una sintaxis inspirada en Python y Perl, con características de programación orientada a objetos: todos los tipos de datos son un objeto, lo que significa que se les pueden asignar propiedades y acciones a toda información y código.

Comparte también funcionalidad con otros lenguajes de programación como Lisp, Lua, Dylan y CLU. Ruby es un **lenguaje de programación interpretado** y su implementación oficial es distribuida bajo licencia de software libre. Sigue el "principio de la menor sorpresa", lo que significa que el lenguaje debe comportarse de tal manera que minimice la confusión de los usuarios experimentados, simplificando el trabajo de programación.

Su desarrollo incorpora partes de diferentes lenguajes: Perl, Smalltalk, Eiffel y Ada, para formar un nuevo lenguaje que incorporara tanto la **programación funcional** (un paradigma de programación declarativa basado en la utilización de funciones aritméticas) como la **programación imperativa** (un conjunto de instrucciones que le indican a la computadora cómo realizar una tarea). Ruby ha sido descrito como un lenguaje de programación **multiparadigma**, lo que, como hemos mencionado, significa que es compatible con otros lenguajes.

Su sintaxis es similar a la de Perl o Python, pero con la ventaja de ser más flexible, ya que permite a sus usuarios alterarlo libremente. Las partes esenciales de Ruby pueden ser quitadas o redefinidas: se puede agregar una funcionalidad a partes ya existentes e incluso puede hacerse en cualquier otro lenguaje. Esta básicamente orientado a **no restringir al desarrollador**.





Este lenguaje cuenta con miles de seguidores en todo el mundo, por ser totalmente libre y además por la popularidad alcanzada por sus aplicaciones, particularmente por la plataforma con la que cuenta para llevar adelante **desarrollos web** con un mínimo de configuraciones.

Si quieren conocer qué piensan los usuarios de este lenguaje de programación, pueden visitar su página [www.ruby-lang.org](http://www.ruby-lang.org), en la que encontrarán enlaces a manuales, tutoriales y referencias que les serán útiles a la hora de iniciarse en Ruby.

## Scratch

La tendencia a nivel mundial es el desarrollo de lenguajes de programación cada vez más accesibles, fáciles de implementar y que potencien la cultura participativa, abierta y libre.

Los impulsores de este movimiento sostienen que hace falta una mayor democratización de estos lenguajes para que cada usuario pueda **adaptar los programas** a sus propias necesidades. Pero para que esto sea posible es necesario ampliar el acceso a las tecnologías y lograr una mayor alfabetización en materia de programación.

Ellos aseguran que al igual que las matemáticas o los idiomas, los lenguajes de programación también deben ser una asignatura más en el currículo educativo: con ese espíritu nació el lenguaje de programación Scratch.

Scratch fue concebido como una **herramienta didáctica** destinada a enseñar programación en particular e informática en general a los alumnos de nivel primario y secundario.

Es una aplicación informática que permite que los niños exploren y experimenten con los conceptos de programación de computadoras mediante el uso de una sencilla interfaz gráfica.

Si bien es aún un desarrollo muy joven –apareció en el 2007–, ha tenido una gran aceptación. Fue desarrollado por The Lifelong Kindergarten group, un área dedicada a los proyectos educativos del MIT, el Instituto de Tecnología de Massachussets, en los Estados Unidos.

Los desarrolladores de este lenguaje entendieron que con la expansión de las computadoras y sus aplicaciones los alumnos se interesarían –cada vez en edades más tempranas– en la **programación como medio creador de diferentes utilidades, aplicaciones y juegos**.

Con Scratch, prácticamente todo se lleva a cabo arrastrando y soltando elementos con el mouse y modificando con el teclado únicamente valores numéricos o textos.





Como podrán intuir es un **software libre**, que se puede instalar y redistribuir libremente en cualquier computadora sin importar cuál sea su sistema operativo: Windows, Mac OS X o Linux. Si bien es un proyecto de código abierto, su desarrollo es cerrado: el código fuente se consigue libremente pero la aplicación se desarrolla en el MIT.

El nombre de Scratch se deriva de la técnica de *scratching* (arañar), una técnica utilizada por los DJs de hip-hop y música electrónica, que consiste en mover un disco de vinilo hacia adelante y hacia atrás sobre el plato del tocadiscos para crear un efecto parecido al de rayar el disco, para construir ritmos y frases melódicas.

La similitud con el scratching musical es la fácil reutilización de piezas: en Scratch todos los objetos, gráficos, sonidos y secuencias de comandos pueden ser fácilmente importados a un nuevo programa y combinados, permitiendo a los principiantes conseguir resultados rápidos y motivándolos a nuevos intentos.

Scratch se utiliza en muchos países y en entornos diferentes, pero ha sido especialmente pensado para escuelas, museos, centros comunitarios y hogares.

El **público** al que está dirigido comprende la franja de los **6 a los 16 años de edad**, pero es también utilizado por usuarios de otras edades por sus características amigables que facilitan la creación de **historias interactivas, animaciones, juegos, música y arte, para publicar en la Web**. Cuenta para ello con numerosos elementos disponibles: imágenes, sonidos y textos.

También permite incorporar las propias grabaciones o archivos de los usuarios. En el caso de los sonidos, cuenta con una grabadora incorporada en el entorno para poder registrarlo y luego utilizarlos para crear nuevos.

Scratch potencia el desarrollo del **pensamiento creativo** y del **trabajo en equipo**, ya que favorece el intercambio entre usuarios estimulando el aprendizaje colaborativo.

### Algunas de las ventajas de Scratch son:

- Se encuentra disponible en distintos idiomas, entre ellos el español.
- La interfaz gráfica permite crear y utilizar un escenario con múltiples fondos y objetos móviles programables.
- En lugar de escribir instrucciones (códigos), permite programar con el mouse, arrastrando bloques autoencajables.
- Posibilita explorar principios básicos de programación sin las complicaciones de sintaxis que tienen otros lenguajes.
- A medida que los usuarios crean sus programas, aprenden el manejo de conceptos como: control de flujo, iteración (repetición o ciclos), condicionales, procedimientos, hilos múltiples y eventos.





- Permite controlar y mezclar diferentes formatos (gráficas, texto, sonido).
- Facilita la manipulación de objetos.

Para empezar a trabajar en Scratch solo hace falta ingresar en su página, suscribirse a la comunidad –que ya cuenta con más de medio millón de usuarios– y ponerse a investigar. Pueden crear sus propios proyectos o intervenir en alguno de los más de mil que se encuentran en desarrollo. Antes de hacerlo, les sugerimos leer la guía de inicio.

En el sitio puede encontrarse una sección exclusiva para educadores, con herramientas, artículos de interés y material educativo para iniciarse en este lenguaje y sus posibilidades.

En este fascículo comprobamos que el mundo del software libre es muy amplio y cada vez son más los programadores y usuarios que se inclinan por este, ofreciendo y compartiendo su conocimiento.

El sitio Black Duck KnowledgeBase recopila la información de más de 200 mil proyectos de código abierto desde el año 2002, y recientemente hizo público un análisis donde muestra cuáles son los lenguajes más populares dentro de la comunidad de software libre. Tres de los que hemos desarrollado en este fascículo se ubican entre los diez primeros lugares.

<b>1</b>	C
<b>2</b>	C++
<b>3</b>	Java
<b>4</b>	Shell
<b>5</b>	<b>Javascript</b>
<b>6</b>	<b>PHP</b>
<b>7</b>	Perl
<b>8</b>	<b>Python</b>
<b>9</b>	SQL
<b>10</b>	C#

Para hacer un análisis correcto de estos diez primeros puestos debemos tener en cuenta la antigüedad de cada uno de los lenguajes. Sin duda los más nuevos deberán contar con más tiempo de pruebas, de desarrollo y también de difusión.

Falta el voto de ustedes, ¡a ver por cuál se inclinan!





## Fuentes

- [www.wikipedia.org](http://www.wikipedia.org)



**Autora:** María Lorena Suárez  
**Coordinación editorial:** Mara Mobilia

