

# ¿Qué puede hacer el software R para resolver tus problemas?

*José Luis Ángel Rodríguez Silva*

## Resumen

R es una herramienta informática (específicamente, un lenguaje computacional) sumamente potente para realizar distintos cálculos científicos, numéricos y estadísticos, así como para crear gráficas y figuras de gran calidad. R, de hecho, tiene hoy en día gran popularidad entre la comunidad académica y entre distintos profesionales. Esto no es casual puesto que R es un excelente producto que, entre una gama de virtudes, es gratuito, relativamente fácil de operar y cuenta con una gran comunidad de internet que contribuye a resolver dudas y problemas, sin costo alguno. La finalidad de este artículo es mostrar algunas de las principales características de R y dar una breve historia de él, mencionar algunas de sus aplicaciones y comentar acerca de su posicionamiento relativo respecto a otras herramientas informáticas similares. Finalmente, se invita al lector al uso de R para que sea una útil herramienta para resolver sus propios problemas.

**Palabras clave:** R, comparación de software, cómputo científico, resolución de problemas, software libre.

## ***WHAT CAN DO THE R SOFTWARE TO SOLVE YOUR PROBLEMS?***

## Abstract

R is a computer tool (specifically, a computer language) extremely powerful for scientific, numerical and statistical calculations, as well as to create high quality graphics and figures. As a matter of fact, R is very popular among academics and several professionals. This is not a coincidence since R is an excellent product because, among other reasons, has no cost, is easy to operate and has a large internet-based community who contributes freely to solve R doubts and problems. The purpose of this article is to show some of the main characteristics of R, recount a brief history, point out some of its applications and comment about its relative positioning respect to other similar computer tools. Finally, we invite the reader to use R to solve his own problems.

**Keywords:** R, software comparison, scientific computing, problem solving, free software.

DOI: <http://doi.org/10.22201/codeic.16076079e.2019.v20n3.a5>



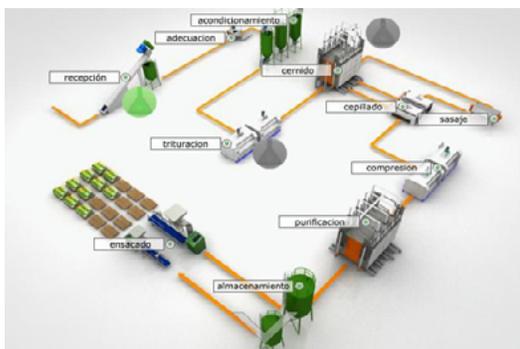
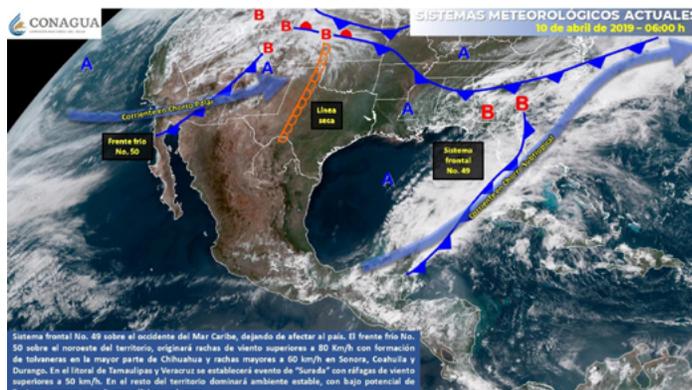
José Luis Ángel Rodríguez Silva

[angel.rodriguez@upa.edu.mx](mailto:angel.rodriguez@upa.edu.mx)

Catedrático del Departamento de Ciencias Básicas, Universidad Politécnica de Aguascalientes. Dr. en Matemática Educativa por parte del Instituto Politécnico Nacional y Dr. en Planeación y Liderazgo Educativo por parte de la Universidad del Valle de México, Campus Santa Fe.

## Introducción y bosquejo histórico

Los lenguajes de programación, formas que tenemos los seres humanos de comunicarnos con las computadoras, son extraordinariamente importantes para la resolución de problemas que se encuentran en casi cualquier ámbito de estudio, aplicación e interés de las actividades humanas. Nos ayudan a resolver problemáticas tan diversas como la determinación del pronóstico del tiempo, la detección de enfermedades y la simulación de procesos industriales. En la figura 1 se pueden apreciar algunas de sus aplicaciones.



**Figura 1.** Algunas actividades humanas de uso de lenguajes de programación.

Fuentes: [Pronóstico Meteorológico General](#), [123rf](#) y [virtualplat.net](#).

## Historia del lenguaje R

R fue creado hace relativamente poco, en 1993, por Robert Gentleman (doctor en estadística por parte de la Universidad de Washington y quien actualmente trabaja en Genentech, Estados Unidos) y por Ross Ihaka (doctor por la Universidad de Berkeley y quien trabaja hoy en día como profesor asociado de la Universidad de Auckland, Nueva Zelanda). Véase la figura 2.



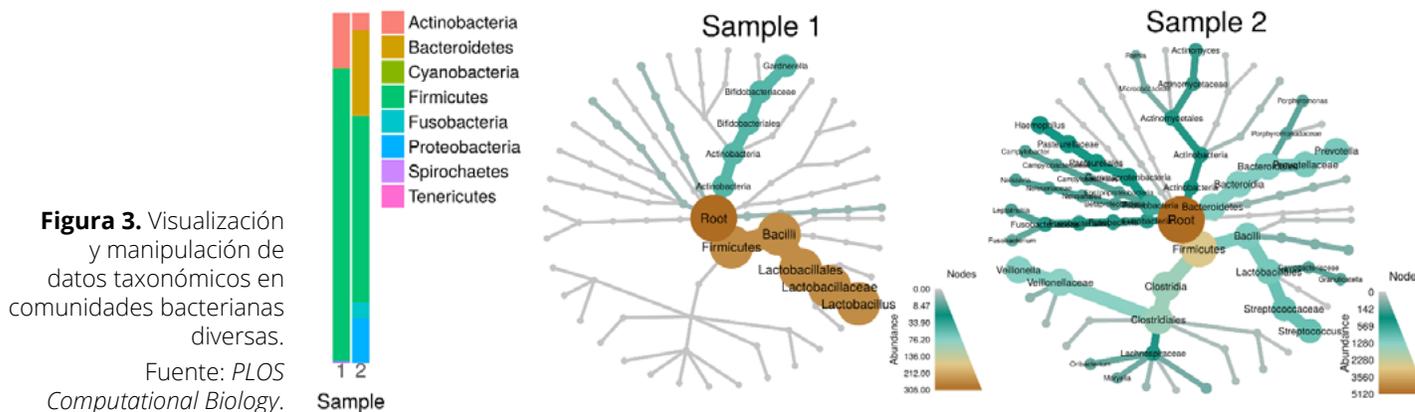
Robert Gentleman



Ross Ihaka

**Figura 2.** Los creadores de R: Robert Gentleman y Ross Ihaka.  
Fuentes: [Computer Hope](#) y [Ross Ihaka's Home Page](#).

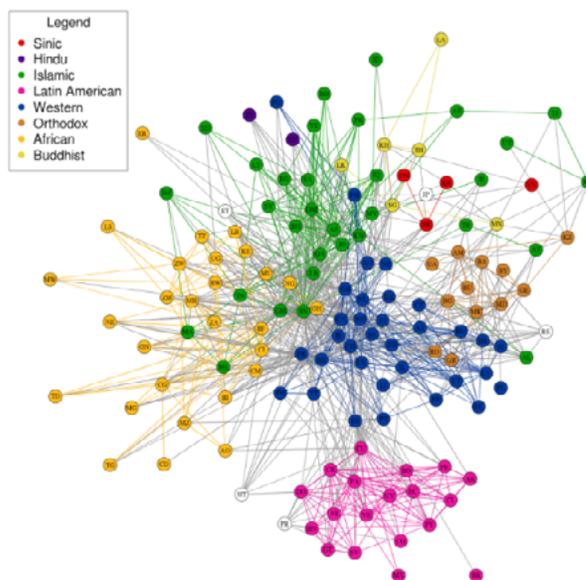
R nació con una fuerte influencia de dos lenguajes: el S, creado por Becker, Chambers y Wilks; y el Scheme, creado por Sussman; en tal sentido tiene una apariencia muy similar a S, aunque en su implementación y semántica es más parecido a Scheme. Además, el objetivo de desarrollar R fue contar con una plataforma libre y gratuita para el uso y desarrollo de diversos algoritmos numéricos y estadísticos. R ha resultado tener una impresionante acogida por parte de la comunidad estadística y científica, y hoy en día sus numerosos usuarios no son únicamente profesionales en aplicaciones cuantitativas, sino que R tiene un amplio margen de utilización que va desde la ciencia a la economía y de la medicina a la ingeniería, e incluso en estudios históricos, legales y sociológicos. Además de sus robustas características de cálculo numérico, R permite la creación de figuras y gráficos de muy alta calidad y con efectos dinámicos, mismo que lo hace sumamente atractivo y, por lo tanto, logra que la labor de modelación, tan importante en muchas ramas de la ciencia, se vuelva más grata y estimulante. Por ejemplo, para un biólogo que estuviera interesado en analizar datos taxonómicos, se podría generar análisis que derivan en imágenes como la que se muestra en la figura 3.



O bien, si nos interesara conocer acerca de cómo se manifiestan los flujos de correos electrónicos entre usuarios de distintos países y a pesar de que se pudiera pensar que tales flujos pudieran ser aproximadamente los mismos entre éstos, la figura 4, hecha con R, nos muestra que los correos electrónicos tienden a ser mucho más fluidos entre países con ciertas afinidades económicas y culturales.

**Figura 4.** Flujos de correos electrónicos entre países.

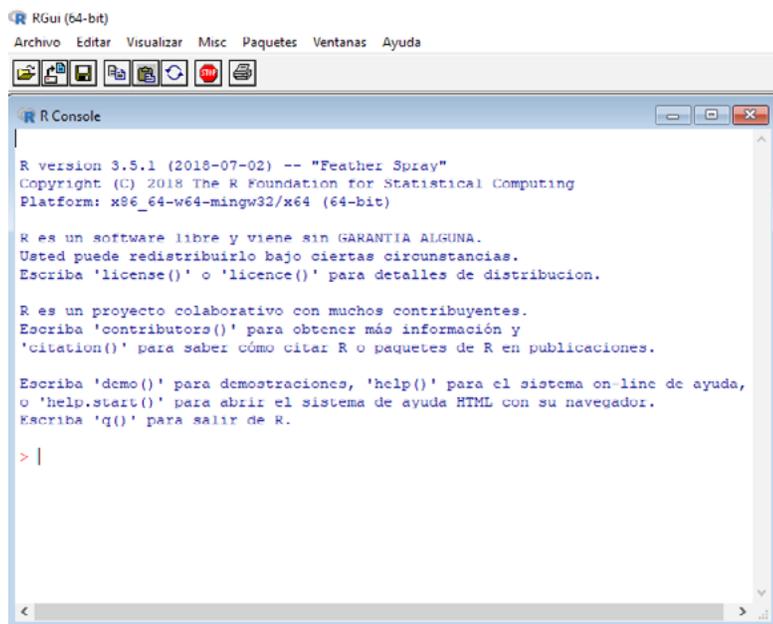
Fuente: [cytadine](http://cytadine.com).



## Descarga de R y ejemplos de aplicación

R puede ser descargado de manera gratuita [aquí](#). Al ingresar a esta página, se selecciona la opción CRAN, que es el acrónimo de *Comprehensive R Archive Network*. En la nueva página se ingresa al servidor que sea de nuestra preferencia, y de ahí, dependiendo del sistema operativo con el cual se esté trabajando (Linux, MacOS o Windows), se toma el programa “base” el cual descarga e instala R en nuestros equipos de cómputo (al usuario principiante se recomienda aceptar los *defaults* al instalar el programa).

Pensando en un entorno Windows, después de que ha sido instalado R y se abre el ícono de acceso directo, se tendrá una ventana de inicio como la que se muestra en la figura 5.



**Figura 5.** Ventana principal de R en Windows. Fuente: Ejecución de R versión 3.5.1

El ícono en rojo es el indicativo de que podemos comenzar a escribir el código que deseamos. Visto así, R puede ser muy bien considerado como una supercalculadora científica. Por ejemplo, si quisiéramos generar 1000 números aleatorios (de una distribución uniforme entre 0 y 1), conocer las principales estadísticas de esos números y generar algunas gráficas de interés como el histograma (figura 6), puede considerarse el siguiente código (el lector puede hacer un proceso de copiado y pegado directamente sobre el ícono en rojo):

```
a <- runif(1000,0,1)
```

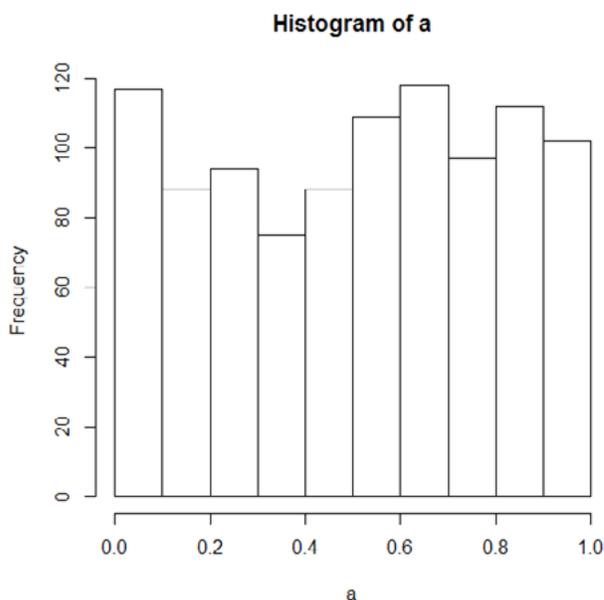
```
summary(a)
```

```
hist(a)
```

El resultado es el siguiente:

```
Min. 1st Qu. Median Mean 3rd Qu. Max.
```

```
0.0004439 0.2407000 0.4847000 0.4932000 0.7388000 0.9985000
```



**Figura 6.** Histograma de un conjunto de datos aleatorios.

Fuente: ejecución de R versión 3.5.1.

¡Con este pequeñísimo código hemos sido capaces de generar una gran cantidad de información, la cual, antes, nos habríamos tardado mucho tiempo en obtener!

Un ejemplo adicional, en el cual se utiliza R ya no tanto como una supercalculadora científica (esto es, por líneas de comandos), sino como un lenguaje de programación, sería el generar un código para resolver un problema típico de álgebra de secundaria, que consiste en encontrar las soluciones (también llamadas “raíces” o “ceros”) de una ecuación cuadrática. Para esto, en la ventana principal de R dentro de “Archivo” seleccionamos “Nuevo script”, que es una ventana de texto en el cual podemos generar código permanente, si así lo queremos. Consideremos el siguiente código que puede encontrar las raíces

---

de una ecuación cuadrática  $ax^2+bx+c=0$ , el cual lo ejemplificamos con la ecuación concreta  $x^2+2x+1=0$ .

```
cuadratica <- function(a,b,c){  
  discriminante <- b^2-4*a*c  
  x1 = (-b + sqrt(as.complex(discriminante)))/(2*a)  
  x2 = (-b - sqrt(as.complex(discriminante)))/(2*a)  
  raices <- c(x1,x2)  
  raices  
}  
cuadratica(1,5,6)
```

El resultado es:

```
[1] -2+0i -3+0i
```

Así, la ecuación tiene dos raíces reales iguales a -2 y -3.

El lector interesado puede consultar a mayor detalle las potencialidades de R mediante la ejecución del comando “demo”. Por ejemplo, para conocer más acerca de la forma de operar de R en gráficos, podemos ejecutar los comandos:

```
demo(graphics)
```

```
demo(persp)
```

Para un aprendizaje a mayor profundidad, existe una gran cantidad de recursos disponibles tanto en la página oficial de R (<https://www.r-project.org/>) como en diversos sitios de internet. Uno de los documentos más atractivos y didácticos es *The R Book* (Crawley, 2013) y en español una buena introducción a R es la de Venables y Smith (2000).

## Consejos de elección de un lenguaje de programación y razones para aprender R

La elección para aprender un lenguaje de programación debe ser meditada y no ser tomada a la ligera, puesto que es grande la inversión de tiempo y pueden, bajo una mala elección, dilapidarse valiosos recursos. Algunos de los aspectos más relevantes a tomarse en consideración son:

- **Ámbito de aplicación.** Aquí debemos preguntarnos qué tipo de aplicación nos interesa resolver. ¿Nuestros problemas son de tipo financiero, ingenieril, de diseño, científico? El respondernos esta pregunta cierra de manera importante la gama de opciones que tenemos.

- 
- **Costo.** El factor económico es también crucial en nuestra decisión. ¿Estamos en condiciones de poder absorber el costo de un determinado producto informático? En este sentido, debe valorarse si en verdad vale la pena invertir nuestros recursos económicos (ciertamente escasos) en tal o cual producto comercial, sobre todo si existe una opción gratuita o libre.
  - **Características propias del lenguaje o entorno de programación.** Existe una gran variedad de entornos de desarrollo informático, desde los que parten de una versión de consola (como es el caso de R), hasta programas muy sofisticados visualmente. ¿Qué es realmente lo que necesitamos? ¿Requerimos un entorno de alto impacto visual en el desarrollo de un programa o basta con tener una forma más sencilla de crear nuestros códigos y obtener los resultados que buscamos?
  - **Características de la máquina o recursos materiales que tenemos.** En este sentido, antes de optar por uno u otro lenguaje, debemos de preguntarnos: ¿las características de nuestra máquina soportan los requerimientos mínimos de dicho lenguaje? Hoy en día, debido a lo robusto que son muchas máquinas y a su bajo costo, esto no parece ser un aspecto tan fuerte para tomar en cuenta, pero de cualquier manera vale la pena tenerlo presente, por seguridad.
  - **Soporte.** Con frecuencia el software comercial ofrece soporte por parte de la propia empresa, el cual puede ayudarnos a solucionar cierto tipo de problemas que pudieran presentarse con su uso. Sin embargo, si bien es cierto que en muchos lenguajes libres o gratuitos no existe de manera obligatoria tal tipo de soporte, en proyectos informáticos importantes como R, hay una impresionante comunidad de expertos que brindan, en distintos foros en internet y sin afán de lucro, ayuda desinteresada y de alta calidad. A su vez, existen extensos repositorios de preguntas y problemas que han tenido otros usuarios, con lo cual es muy probable que podamos encontrar respuestas a problemas iguales o similares al de nosotros.
  - **Disponibilidad de bibliotecas especializadas.** Un punto importante a considerar en la elección de una plataforma informática es si existen paquetes especializados (conjunto de programas ya preescritos) para resolver ciertos problemas específicos. Este es ciertamente el caso de R en el cual, al momento de escribir estas líneas, contaba con 14,057 paquetes para resolver problemas en áreas tan variadas como genética, finanzas, matemáticas financieras, proyectos de ingeniería, etcétera. Cada uno de estos paquetes, en otros entornos, podría tener costos muy altos, mientras que en R son gratuitos y de libre acceso.
  - **Paradigma de visualización y desarrollo.** De manera muy general, existen dos grandes tipos o paradigmas de programación. Por un lado, los del tipo *wysiwyg* (*What You See Is What You Get*, o bien, Lo Que Usted Ve Es Lo Que Obtiene), que es sobre el cual se desarrollan productos como Excel, Minitab, SPSS y Statistica, para hablar sólo de algunos de los productos estadísticos informáticos más relevantes. En ellos, salvo aplicaciones muy especializadas no existe la necesidad de programar. Por otro lado, están los del tipo que requiere programación. En este último caso, tal y como lo reconocerán aquellos usuarios que hayan tenido la necesidad de llevar a cabo cierto tipo de programación, se requiere de tiempo para madurar habilidades de diseño conceptual en materia de programación, reconocer la sintaxis y comandos más utilizados y esenciales (como la declaración

de variables y el uso de bucles y condicionales), así como el manejo de interfaces de comunicación con otro tipo de programas. Sin embargo, después de pasar lo escarpado de esta “curva de aprendizaje”, se obtienen diversas ganancias; de manera operativa, la más importante es el hecho de estar en condiciones de desarrollar código en el cual uno sabe, de forma precisa y puntual, qué es lo que hace y no hay dudas del tipo “caja negra”, como en los primeros productos comerciales mencionados en este párrafo, en los cuales usualmente no conocemos la forma en que se llevó a cabo la programación de algún determinado algoritmo, puesto que ésta es cerrada y críptica, y siempre dejará ciertas dudas al respecto. Quizá lo más relevante sea el hecho de que la programación realizada por nosotros mismos nos permite modificar a nuestro criterio los programas asociados, expandirlos o acotarlos, según se vayan modificando nuestras necesidades.

Tal y como ha quedado clarificado en los párrafos anteriores, R es un potente lenguaje de programación que junto con S-plus (el cual es comercial y muy similar a R) son, sin lugar a dudas, los dos lenguajes más utilizados en la comunidad de científicos que tienen que hacer uso de la estadística. De manera puntual, las siguientes son algunas de las razones por las cuales aprender R:

- Manejo extraordinariamente eficiente de datos numéricos y estadísticos.
- Programa orientado a objetos.
- Desarrollo de programación estructurada.
- Se ejecuta en los sistemas operativos más comunes: Windows, MacOS y Linux.
- Impresionantes características gráficas.
- Código libre y abierto.
- Gratuito.
- Gran cantidad de librerías especializadas.
- Vasta cantidad de manuales tanto en inglés como en español (además de otros idiomas), tanto a nivel de principiante, intermedio y avanzado.
- Soporte de una comunidad que sin fines de lucro está dispuesta a auxiliarnos en dudas con respecto al desarrollo de programas en R.
- Programa relativamente pequeño (en la última versión, la 3.5.1 para Windows, el peso es de 62 MB, tanto para las máquinas con tecnologías de 32 y 64 bits).

## **Comparación de R con otros lenguajes y entornos de desarrollo estadístico**

Con la finalidad de poder establecer un mejor marco de comparación entre diversas herramientas informáticas, vale la pena llevar a cabo un pequeño comparativo de R con respecto a otros lenguajes similares, orientados hacia el análisis de problemas numéricos y estadísticos. Para tales fines, se tomaron

---

lo que a consideración del autor son los más relevantes tanto en contextos académicos como profesionales. Para llevar a cabo el comparativo, se analizaron características de gratuidad, código libre, existencia de librerías especializadas y soporte de cada uno de los programas. Los hallazgos se presentan en la tabla 1, en la cual el lector puede dar clic en el nombre de cada software para conocer más de él; en algunos casos se permite la descarga con o sin limitantes, siendo la más frecuente el hecho de ser versiones de prueba por tiempo limitado).

Software	¿Gratuito?	¿Código libre?	¿Librerías especializadas?	¿Soporte?
R	😊	😊	😊	😊 gratuito
BMDP	😞	😞	😞	😊 paga
E-Views	😞	😞	😊	😊 paga
Gauss	😞	😞	😊	😊 paga
JMP	😞	😞	😊	😊 paga
Minitab	😞	😞	😊	😊 paga
S-Plus	😞	😞	😊	😊 paga
SAS	😞	😞	😞	😊 paga
SPSS	😞	😞	😊	😊 paga
STATA	😞	😞	😊	😊 paga
Statgraphics	😞	😞	😞	😊 paga
Statistica	😞	😞	😊	😊 paga
Excel	😞	😞	😞	😊 paga
Matlab	😞	😞	😊	😊 paga
Mathematica	😞	😞	😊	😊 paga
Scilab	😊	😊	😞	😞 NA

Tabla 1. Comparativo de las características didácticas de diversas herramientas informáticas.

Nota: la simbología 😊 significa que el software bajo consideración goza de la propiedad señalada en el inicio de la columna, mientras que si tiene el símbolo 😞, no tiene dicha característica. Así, por ejemplo, el E-Views no es gratuito (😞) pero si tiene librerías especializadas (😊).

## Conclusiones

R es un lenguaje y entorno de programación científica, numérica y estadística de gran impacto y de enormes potencialidades. Si bien es cierto que hay un precio por aprender el lenguaje (igual que en el aprendizaje de casi cualquier otra plataforma de características similares), vale la pena el esfuerzo por la simplicidad, elegancia y compacidad del código que se genera, por la gran cantidad de bibliotecas especializadas y por el excelente soporte que brindan de manera generosa diversos usuarios expertos de este programa. Sin lugar a dudas, se recomienda ampliamente su utilización.

---

## Para conocer más

- ❖ Brown, C. (2009). *Reasons to learn the R Programming Language* (duración, 7:28 minutos). Recuperado de <http://www.youtube.com/watch?v=W2GZFeYGU3s>.
- ❖ Crawley, D. (2013). *The R book*. John Wiley & Sons: United Kingdom. Recuperado de: <https://www.cs.upc.edu/~robert/teaching/estadistica/TheRBook.pdf>.
- ❖ Ferguson, A. (2004). A History of Computer Programming Languages. Recuperado de: [http://cs.brown.edu/~adf/programming\\_languages.html](http://cs.brown.edu/~adf/programming_languages.html).
- ❖ Ingeniería de Software. (2018). Lenguajes de programación más populares. Recuperado de: <https://ingenieriadesoftware.es/lenguajes-programacion-populares-todos-rankings/>.
- ❖ La Revista Informática.com. (2006). *Historia de los lenguajes de programación*. Recuperado de: <http://www.larevistainformatica.com/historia-lenguajes-programacion.htm>.
- ❖ Revolution Analytics. (2013). *R: The most powerful and most widely used statistical software* (duración 1:33 minutos). Recuperado de: <https://www.youtube.com/watch?v=y-HOQnMsYL>.
- ❖ The R Project for Computing Statistics. (2019). Recuperado de: <https://www.r-project.org/>.
- ❖ Venables, W. y Smith, D. (2000). Introducción a R: Notas sobre R: Un entorno de programación para Análisis de Datos y Gráficos. Recuperado de: <http://cran.r-project.org/doc/contrib/R-intro-1.1.0-espanol.1.pdf>.

## Cómo citar este artículo

- ❖ Rodríguez Silva, José Luis Ángel (2019). ¿Qué puede hacer el software R para resolver tus problemas? *Revista Digital Universitaria* (RDU). Vol. 20, núm. 3 mayo-junio. DOI: <http://doi.org/10.22201/codeic.16076079e.2019.v20n3.a5>.

Recepción: 06/01/17 Aprobación: 21/03/19