

¿QUÉ ES UN LENGUAJE DE PROGRAMACIÓN?

Es el método que se utiliza para dar instrucciones a los diferentes componentes del ordenador. A través de este lenguaje se crean algoritmos que van desarrollando diferentes funcionalidades, creando un conjunto que se convierte en aplicaciones, sitios web, programas (Facebook, Netflix, Whatsapp, Youtube).



El lenguaje básico está representado por un conjunto de bits, “unos y ceros”, que al combinarse dan significado a las instrucciones, se conoce como código binario. Este lenguaje básico es bastante complejo, por lo que se han ido creando diferentes lenguajes de programación, para proporcionar las instrucciones de manera más sencilla e inteligible.

Todo lenguaje de programación tiene una *sintaxis*; esta se refiere a las reglas de escritura del código, así como el conjunto de símbolos que utiliza.

La *semántica* se refiere al contenido del código, son las reglas para transformar comandos en instrucciones lógicas.

La *pragmática*, se refiere al contexto del código, así como a la eficacia del mismo.

Los lenguajes de programación se clasifican:

- Lenguajes de bajo nivel: son lenguajes ligados al hardware, no suelen poder transportarse de un ordenador a otro, actualmente son poco usados. En ellos se incluyen el *lenguaje máquina*, compuesto por ceros y unos (código binario), y el *lenguaje ensamblador*, que son un pequeño conjunto de instrucciones que se ejecutan de forma secuencial, para ser entendidos por la máquina los textos deben ser transformados por un programa ensamblador.
- Lenguajes de alto nivel: más cercanos al lenguaje humano, se trata de una serie de instrucciones, reglas y operadores que permiten crear programas.
- Lenguajes interpretados: son aquellos que una vez creados pueden ser interpretados por cualquier sistema. Son algo más lentos ya que la traducción ha de realizarse cada vez que el programa se ejecuta. (Java, Python, R).
- Lenguajes compilados: deben ser compilados (traducidos), para poder ser ejecutados, solo han de compilarse una vez, y posteriormente se ejecutarán de forma automática, por ello son más rápidos.

INTRODUCCIÓN AL LENGUAJE “R”

R es un entorno computacional integrado, especialmente diseñado para el análisis de datos, los cálculos estadísticos y las representaciones gráficas. Es un lenguaje de programación sencillo. Fue desarrollado por Robert Gentleman y Ross Ihaka, en el departamento de Estadística de la Universidad de Auckland, en 1993, basándose en el lenguaje de programación S, creado en 1970. R es un software GNU (General Public License), es un conjunto de programas de uso libre y gratuito.

R es un lenguaje de programación interpretado, esto quiere decir que las instrucciones son interpretadas directamente sin que intervenga ningún software externo. A diferencia de otros softwares, R utiliza “objetos” (vectores, matrices, gráficos), como unidad básica. Los diferentes objetos pertenecen a diferentes clases, por tanto las funciones se comportarán de manera distinta, dependiendo de la clase a la que pertenezca el objeto. El lenguaje R hace posible coordinar diferentes funciones estadísticas en un solo programa.

INSTALAR “R”

Accederemos a la dirección web: <http://cran.es.r-project.org>, en la que nos proporcionan el enlace necesario para descargar e instalar el programa en función del sistema operativo que utilicemos.

- R for Linux (Debian, Fedora/Redhat, Ubuntu)
- R for macOS
- R for Windows

UTILIDADES Y FUNCIONALIDADES DEL LENGUAJE “R”

R está diseñado específicamente para el análisis estadístico y la manipulación de datos, se aplica en estudios que necesitan un análisis profundo, ya que recopila, gestiona y almacena diferentes tipos de datos, ya sean numéricos o no.

Con R se pueden ejecutar códigos de saltos condicionales, entrada y salida de datos, bucles, realizar cálculos con matrices. Proporciona herramientas para hacer cálculos estadísticos. Nos permite ajustar modelos mediante regresión lineal o regresión logística entre otros.

Cuenta con paquetes específicos para crear gráficos estadísticos, así como otros con algoritmos para el aprendizaje automático.

Es compatible con otros lenguajes de programación como C o Fortran, permitiendo mejorar el análisis de datos.

El lenguaje de programación R es uno de los más utilizados en investigación científica, en ámbitos como el aprendizaje automático, biomedicina, biotecnología, bioinformática o economía.

Las funciones en R nos permiten escribir un fragmento del código y ejecutarlo para distintos datos. Es decir, es como una sub-rutina que resuelve una sub-tarea. A la hora de resolver un problema complicado, se recurre a descomponerlo en sub-problemas menos complicados que serán resueltos por funciones sencillas.

COMANDOS BÁSICOS EN “R”

1. **Introducir el valor en una variable:** las variables podrán ser numéricas o textuales (nombres), en el caso de que sean nombres deben ir entre comillas.

variable = valor

variable <- valor

Para poder ver el valor de la variable debemos introducir: print (variable)

Ejemplos:

```
> num1=6
```

```
num2<-1.2
```

```
name1="perro"
```

```
name2<-“gato”
```

2. **Borrar objetos en memoria:** utilizamos la función rm().

Ejemplo:

rm(x) elimina el objeto x,

rm(x,y) elimina ambos objetos x y

rm(list=ls()) elimina todos los objetos en memoria

3. **Generar datos al azar:** rnorm(n), genera valores aleatorios acorde con los parámetros indicados donde “n” representa el número de valores que deseamos obtener.

Ejemplos:

```
> rnorm (4)
```

```
[1] -0.3455678 0.6498987 -0.2656743 0.8985981
```

```
> rnorm (3)
```

```
[1] 1.7823456 -2.8965844 0.1367856
```

4. **Obtener N números aleatorios en el intervalo [0,1]:** `runif(n)`, donde “n” es el número de resultados que queremos obtener. El resultado será diferente cada vez que ejecutemos la función.

Ejemplos:

```
> runif (4)
```

```
[1] 0.657859467 0.48456453 0.985674536 0.895645638
```

Si volvemos a ejecutar la misma función observamos que el resultado es diferente:

```
> runif (4)
```

```
[1] 0.12345678 0.89786756 0.34243413 0.234367845
```

5. **Listar la información existente en el directorio de trabajo:** `ls ()`.

Podemos listar los objetos cuyo nombre contenga un determinado caracter, utilizando la función `pat`.

Ejemplo:

Si se introduce en el comando `ls`, en este caso se nos mostrarían los valores introducidos en los apartados anteriores:

```
> ls()
```

```
[1] "num1" "num2" "name1" "name2"
```

```
>ls(pat=m)
```

```
[1] "name1" "name2" "name3"
```

6. **Obtener información de la memoria:** `ls.str()`.

Ejemplo:

```
> ls.str()
```

```
num1: num 6
```

```
num2: num 1.2
```

```
name1: "perro"
```

```
name2: "gato"
```

7. Crear un comentario: #Comentario.

Ejemplo:

Este texto no se va a ejecutar

OPERACIONES ARITMÉTICAS

Es importante conocer el orden de preferencia que se aplica en R a la hora de realizar operaciones

1. Potencias
2. Productos y divisiones
3. Sumas y restas

Si existen dos operaciones con el mismo orden de preferencia, se resolverán de izquierda a derecha. El orden de preferencia puede modificarse mediante el uso de paréntesis: aquellas operaciones que aparezcan entre paréntesis tendrán preferencia.

OPERADORES ARITMÉTICOS

Suma	+
Resta	-
Multiplicación	*
División	/
Potencia	^ ó **
Raíz cuadrada	sqrt()

- **SUMA:** asignaremos valor a dos o más variables, y a continuación utilizaremos el operador “+” para proceder a la suma.

Ejemplo: > a=4; b=3

> a+b

[1] 7

- **RESTA:** llevaremos a cabo el mismo proceso que en la suma, pero utilizando el operador “-”.

Ejemplo: > a=4

> b=3

> a-b

[1] 1

- **MULTIPLICACIÓN:** al igual que en los casos anteriores, asignamos valores a las variables y operamos con el operador “*”.

Ejemplo:

> a=4; b=3

> a*b

[1] 12

- **DIVISIÓN:** llevaremos a cabo el mismo proceso pero utilizando el operador “/”.

Ejemplo:

> a=30; b=2

> a/b

[1] 15

- **POTENCIAS:** para calcular potencias asignaremos valores a las variables, y procederemos a operar, en este caso podemos optar por dos operadores “^” o “**”.

Ejemplo:

> a=3; b=6

> a**b

[1] 729

- **RAÍZ CUADRADA:** se puede calcular raíces cuadradas con la función “sqrt()”.

Ejemplo:

> sqrt(16)

[1] 4

· FUNCIONES LOGARÍTMICAS

Logaritmo neperiano	$\log(x)$
Logaritmo en base 10	$\log_{10}(x)$
Logaritmo en base 2	$\log_2(x)$
Logaritmo (cualquier base)	$\log(x, \text{base})$
F. exponencial (base e)	$\exp(x)$

Ejemplo:

> $\log(\exp(17))$

[1] 17

> $\log(81, 3)$

[2] 4

· **FUNCIONES TRIGONOMÉTRICAS:** son muy utilizadas en el área de la ciencia, deberemos tener en cuenta que hay que utilizar los valores en radianes.

Seno	$\sin(x)$
Coseno	$\cos(x)$
Tangente	$\tan(x)$
Arco-seno	$\text{asin}(x)$
Arco-coseno	$\text{acos}(x)$
Arco-tangente	$\text{atan}(x)$

Ejemplo:

> $\sin(0)$

[1] 0

> $\cos(0)$

[2] 1

Aclaración: El contenido de la x debe estar en **radianes**, no en grados

OPERACIONES LÓGICAS

Estos símbolos sirven para crear condiciones. Los resultados de estas operaciones solo pueden ser uno de estos dos valores: VERDADERO o FALSO. Se usan como condiciones en los bucles “while” y también en los “if” condicionales.

Mayor (>)	Mayor o igual (>=)	Igualdad (==)
Menor (<)	Menor o igual (<=)	Desigualdad (!=)

Ejemplo:

> 5>3

[1] FALSE

> 2<=9

CUIDADO: El “=” va siempre después del “<” o del “>”

[2] TRUE

> 8!=8

[3] FALSE

EJERCICIOS PROPUESTOS

EJERCICIO 1: Crear los siguientes objetos, asignándoles el valor que se indica.

Grado1, grado2 grado3, con los valores; odontología, farmacia, biotecnología. Alumnado1 con valor 100, alumnado2 con valor 150, alumnado3 con valor 80.

Solución:

```
grado1="odontología"
```

```
grado2="farmacia"
```

```
grado3="biotecnología"
```

```
alumnado1=100
```

```
alumnado2=150
```

```
alumnado3=80
```

EJERCICIO 2: Generar de forma aleatoria 5 números decimales comprendidos en el intervalo (3,4).

Solución:

```
runif(5,3,4)
```

```
[1] 3.537344 3.629892 3.362016 3.860888 3.930647
```

EJERCICIO 3:

¿Qué comando debo utilizar para borrar una variable?

Solución:

```
rm(variable)
```

EJERCICIO 4:

¿Cómo puedo listar todos los objetos que hay en la memoria? ¿Y objetos que contengan la letra "a"?

Solución:

```
ls()          #Para todos los objetos
```

Is(pat="a") #Para los que tengan "a"

EJERCICIO 5:

Calcula la suma, producto y cociente de 3 y 2. ¿Cuánto sería 3^2 ? ¿Y raíz cuadrada de 3?

Solución:

$3+2$

$3*2$

$3/2$

3^2 #También valdría $3**2$

$\text{sqrt}(3)$

EJERCICIO 6:

Dada una variable "num" ya creada y con un valor numérico asignado por el usuario, calcula el valor de $\sin(e^{\text{num}})$. Calcula también el logaritmo en base 7 del arco-seno de num.

Solución:

$\sin(\exp(\text{num}))$

$\log(\text{asin}(\text{num}))$

EJERCICIO 7:

Dadas dos variables "num1" y "num2" ya creadas y con valores numéricos asignados por el usuario, calcula si num1 es mayor, si es mayor o igual, y si son iguales.

Solución:

$\text{num1} > \text{num2}$

$\text{num1} \geq \text{num2}$

$\text{num1} == \text{num2}$