

Actividad | 1 | Análisis de Conceptos

Métodos Numéricos

Ingeniería en Desarrollo de Software



TUTOR: Miguel Ángel Rodríguez Vega

ALUMNO: Adriana Esteban López

FECHA: 11 de noviembre de 2024

INDICE

Introducción 03

.....

Descripción 04

.....

Justificación 05

.....

Desarrollo 06

.....

Conclusión 13

.....

INTRODUCCIÓN

Los métodos numéricos con técnicas mediante las cuales se pueden formular problemas matemáticos para que estos puedan ser resueltos a través de operaciones aritméticas; siendo el objetivo principal el encontrar soluciones aproximadas a grandes problemas, y es aquí donde entran los métodos numéricos.

Los métodos numéricos en esencia son una serie de operaciones numéricas (repetitivas) para encontrar un valor numérico que nos estará dando como resultado una solución que, aunque no es un valor exacto se le aproxima, lo cual conocemos como un valor aproximado, y para ello podemos utilizar algunos de los siguientes modelos:

1. Teorema del Bolzano: es una hipótesis matemática que es utilizada para realizar gráficos de funciones que son definidas sobre un intervalo de 2 ejes
2. Método de Bisección: es conocido también como de corte binario, de intervalos o de Bolzano, el cual siempre divide la función a la mitad, en cuanto se presenta un cambio de signo sobre un intervalo se evalúa el valor de la función en ese intervalo
3. Método de Newton: este método se deduce a partir de una interpretación geométrica.

Dentro de análisis de métodos numéricos existen conceptos básicos:

1. Aproximación numérica y errores
2. Exactitud y precisión

Los métodos numéricos se implementan mediante un lenguaje de computación, en el transcurso de esta actividad se estará haciendo uso de Rstudio.

DESCRIPCIÓN

Para esta actividad se estará realizando el uso de Rstudio en el cual se estarán ejecutando las indicaciones de acuerdo al siguiente archivo:

umi.edu.mx/coppel/IDS/plataforma/files/programasnr/Valores_numericos.R

Para lo cual se estará plasmando la realización a través de print de pantalla que se estarán plasmando en el apartado de **DESARROLLO** del presente documento.

JUSTIFICACIÓN

Los métodos numéricos pueden ser utilizados en:

1. Cálculo de derivadas
2. Integrales
3. Ecuaciones diferenciales
4. Operaciones con matrices
5. Interpolaciones, etc

Pero la mejor manera de entender los métodos numéricos es aplicándolos directamente en las situaciones que sea necesario para entender cómo funcionan, dicho sea de paso, el uso de los métodos numéricos en la Ingeniería puede ser de gran utilidad para:

1. Dar solución a problemas en los cuales no se puede encontrar una solución analítica
2. Dar solución a problemas que requieren de una gran cantidad de cálculos

Como se mencionó en la Introducción, en el desarrollo de esta actividad se estará haciendo uso de Rstudio para la ejecución de diferentes instrucciones, el uso de este programa es debido a que tiene:

1. Interfaz de fácil uso
2. Es gratuito
3. De fácil sintaxis para su uso

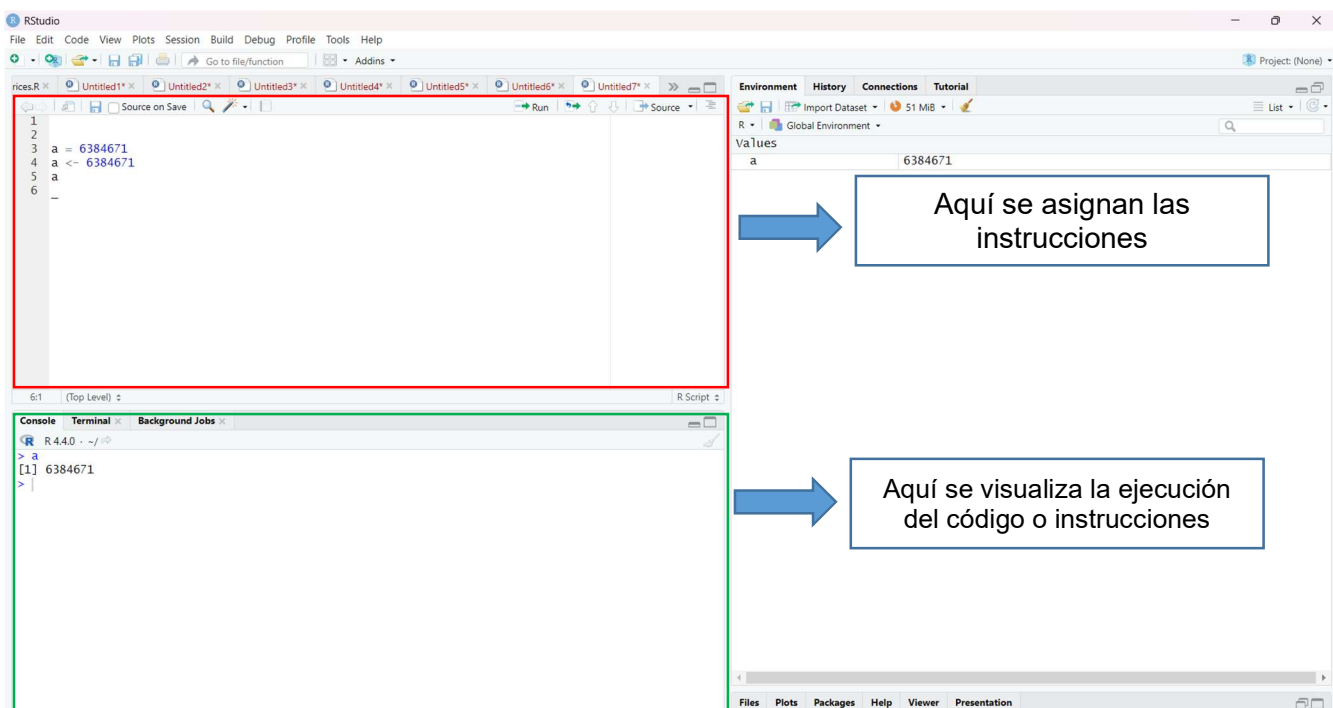
DESARROLLO

En el desarrollo de esta actividad se estará ejecutando cada una de las instrucciones del archivo que se comparte en la Descripción y se estará plasmando el print de pantalla ejecutado en Rstudio.

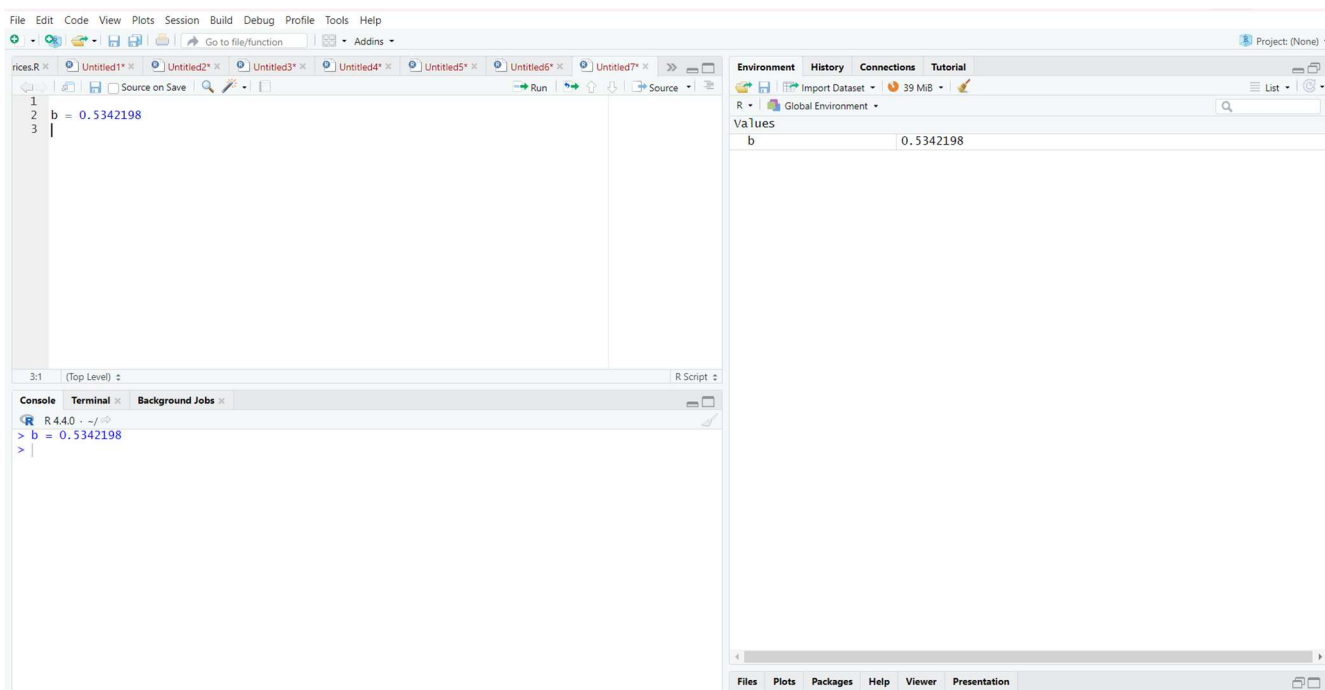
Dígitos Enteros y Fracciones

Para definir que una variable es igual a un determinado valor podemos utilizar los signos `=` o `<=`: en este ejemplo se realizó el uso de la variable **a** y se le asignó el valor de **6384671**

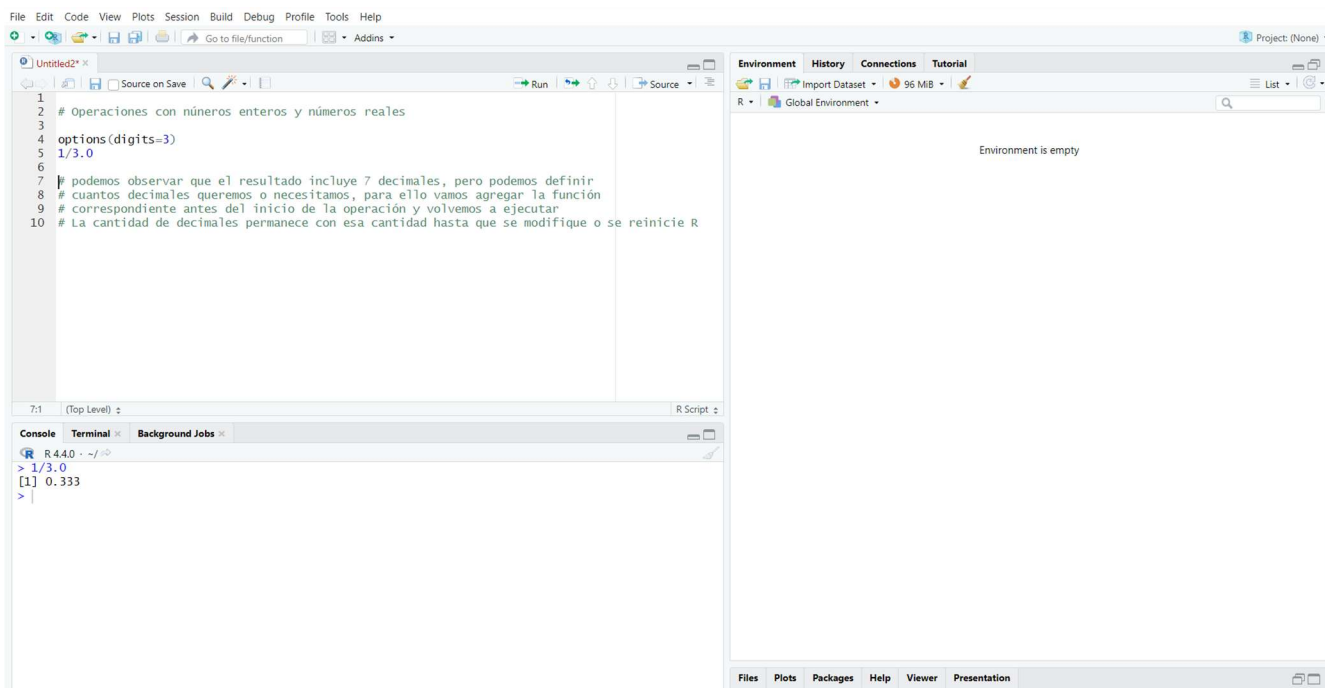
1. Primero escribimos **a = 6384671** ó **a <- 6384671**
2. Luego escribimos **a** y damos un ENTER para visualizar el resultado de la ejecución en la parte inferior de la pantalla



Para asignar un número fraccionario, lo escribimos así: `b <- 0.5342198` y para desplegar el valor damos Enter

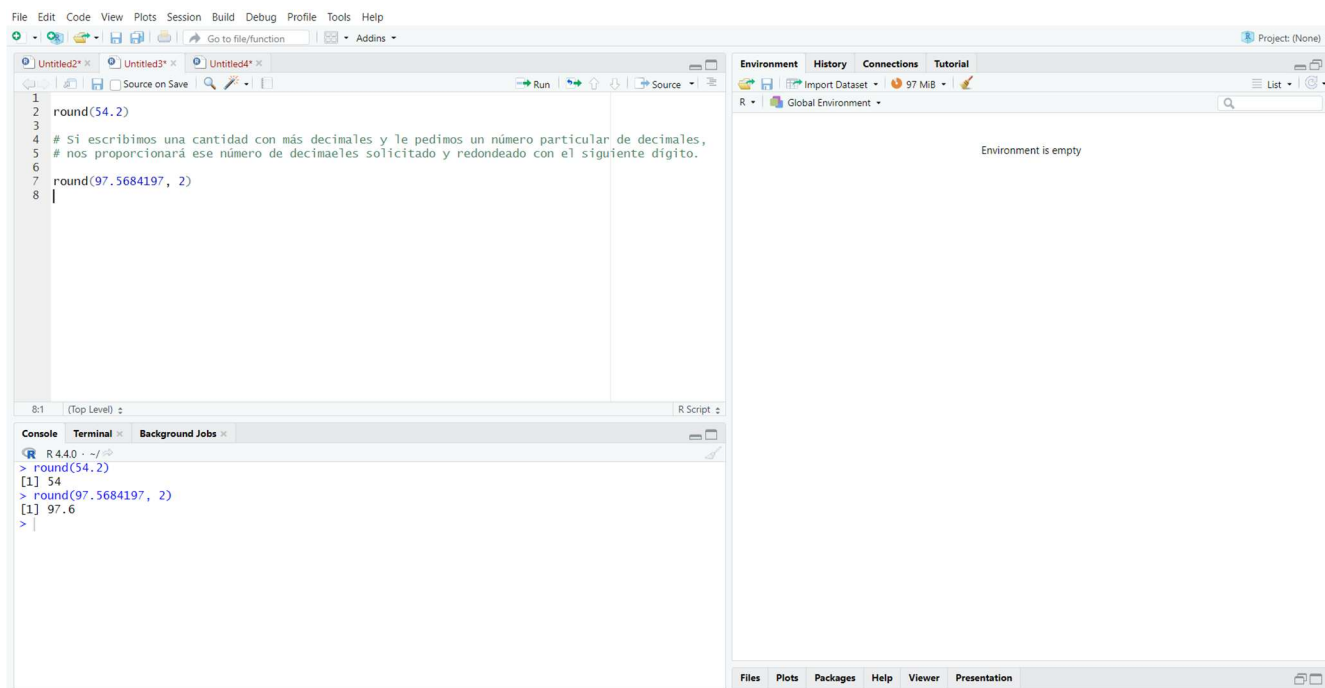


Operación con los valores de un número entero y un valor real (con punto decimal, por ejemplo, la división de 1 entre 3.0; podemos observar que nos devuelve un valor con decimales, sin embargo, esta cantidad de decimales se puede modificar con la función **options** y el modificador **digits = n**, donde **n** es el número de decimales que queremos.



Redondeo, aquí utilizamos la función `round(x,n)`; donde `x` es el valor y `n` es la cantidad de decimales, sin `n` es sin decimales, por ejemplo para el valor de 54.2 el redondeo sería: `round(54.2)`

Para esta función solo se anota tal cual, y se da enter para obtener el resultado, que en este caso es de **54**



The screenshot shows the RStudio interface. The script editor on the left contains the following code:

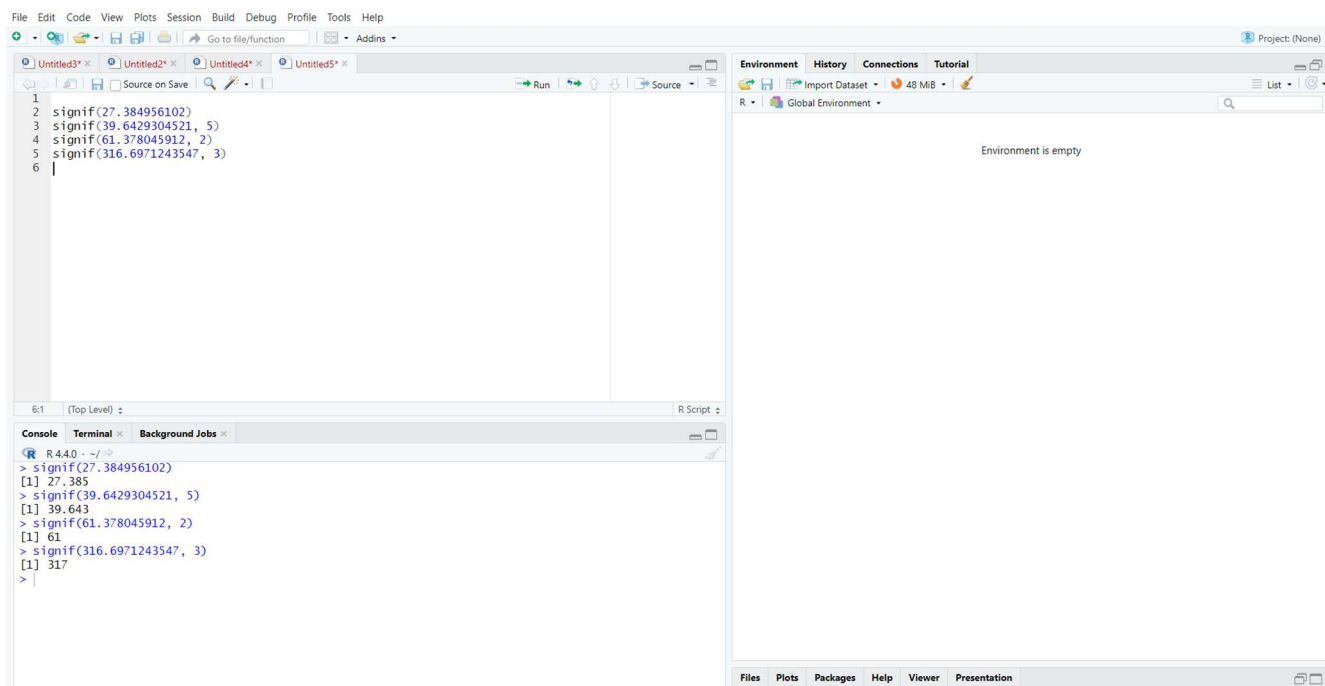
```
1 round(54.2)
2
3
4 # Si escribimos una cantidad con más decimales y le pedimos un número particular de decimales,
5 # nos proporcionará ese número de decimales solicitado y redondeado con el siguiente dígito.
6
7 round(97.5684197, 2)
8
```

The console at the bottom shows the execution results:

```
R 4.4.0 ~ /j
> round(54.2)
[1] 54
> round(97.5684197, 2)
[1] 97.6
>
```

The environment pane on the right is empty, displaying "Environment is empty".

Dígitos significativos con la función `signif(x,n)` en donde se redondea a `x` con `n` dígitos significativos, es decir, que de la cantidad **27.384956102** si la anotamos `signif(27.384956102, 2)` va a redondear la cantidad para mostrar solo 2 dígitos, a continuación se muestran algunos ejemplos:



The screenshot shows the RStudio interface. The script editor on the left contains the following code:

```
1 signif(27.384956102)
2
3 signif(39.6429304521, 5)
4 signif(61.378045912, 2)
5 signif(316.6971243547, 3)
6
```

The console at the bottom shows the execution results:

```
R 4.4.0 ~ /j
> signif(27.384956102)
[1] 27.385
> signif(39.6429304521, 5)
[1] 39.643
> signif(61.378045912, 2)
[1] 61
> signif(316.6971243547, 3)
[1] 317
>
```

The environment pane on the right is empty, displaying "Environment is empty".

Definir variables y asignar valores

`e <- exp(1)` asigna un valor a la variable (Base de los logaritmos naturales $e = 2.718281$), también podemos utilizar el signo igual (=) en lugar de <-

`e` imprime el valor que tenga la variable

Ejemplo 1

```
e <- 2.718281
```

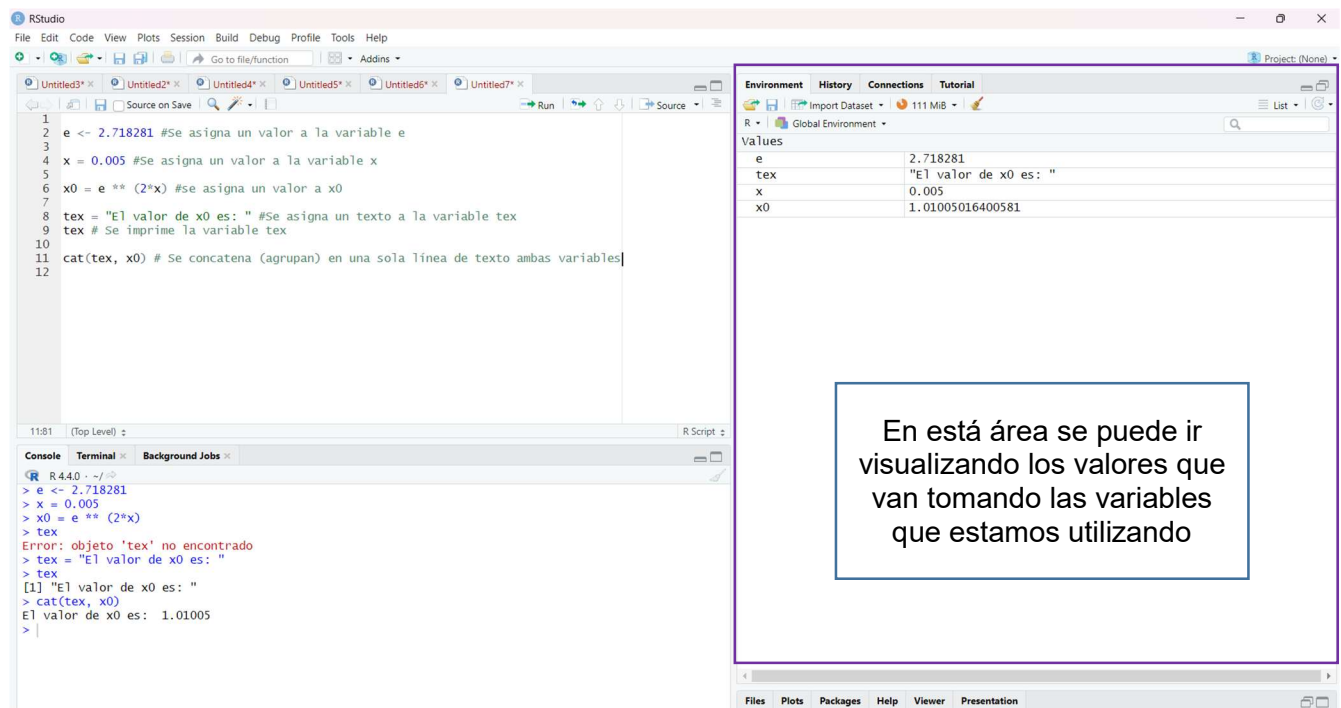
```
x = 0.005
```

```
x0 = e ** (2*x)
```

```
tex = "El valor de x0 es: "
```

```
tex
```

```
cat(tex, x0)
```



The screenshot shows the RStudio interface. The script editor on the left contains the following code:

```
1 e <- 2.718281 #Se asigna un valor a la variable e
2
3 x = 0.005 #Se asigna un valor a la variable x
4
5 x0 = e ** (2*x) #Se asigna un valor a x0
6
7 tex = "El valor de x0 es: " #Se asigna un texto a la variable tex
8 tex # Se imprime la variable tex
9
10 cat(tex, x0) # Se concatena (agrupan) en una sola línea de texto ambas variables
11
12
```

The console at the bottom shows the execution output:

```
> e <- 2.718281
> x = 0.005
> x0 = e ** (2*x)
> tex
Error: objeto 'tex' no encontrado
> tex = "El valor de x0 es: "
> tex
[1] "El valor de x0 es: "
> cat(tex, x0)
El valor de x0 es: 1.01005
>
```

The Environment pane on the right shows the following variables and their values:

Variable	Value
e	2.718281
tex	"El valor de x0 es: "
x	0.005
x0	1.01005016400581

A text box with a blue border is overlaid on the Environment pane, containing the text: "En esta área se puede ir visualizando los valores que van tomando las variables que estamos utilizando".

Ejemplo 2:

```
x0 = 1
```

```
x1 = (x0 - pi) * (x0 + 1)
```

```
x1
```

```
cat("x0 = ", x0, "\n", "x1 = ", x1)
```

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Run Source

```
1 x0 = 1 #Se asigna el valor de 1 a la variable x0
2 x0 #se imprime el valor de x0
3 x1 = (x0-pi) * (x0+1) #se establece que la variable x1 es igual a la operación que se indica
4 x1 # Se imprime el valor de x1
5
6
7 cat("x0 =", x0, "\n", "x1 =", x1) # la función cat sirve para concatenar y mostrar los objetos
8 # en pantalla
9
```

Environment History Connections Tutorial

R • Global Environment • 94 MB

Values

x0	1
x1	-4.28318530717959

Console Terminal Background Jobs

R 4.4.0 ~ /

```
> x1
[1] -4.283185
> cat("x0 =", x0, "\n", "x1 =", x1)
x0 = 1
x1 = -4.283185
> |
```

Files Plots Packages Help Viewer Presentation

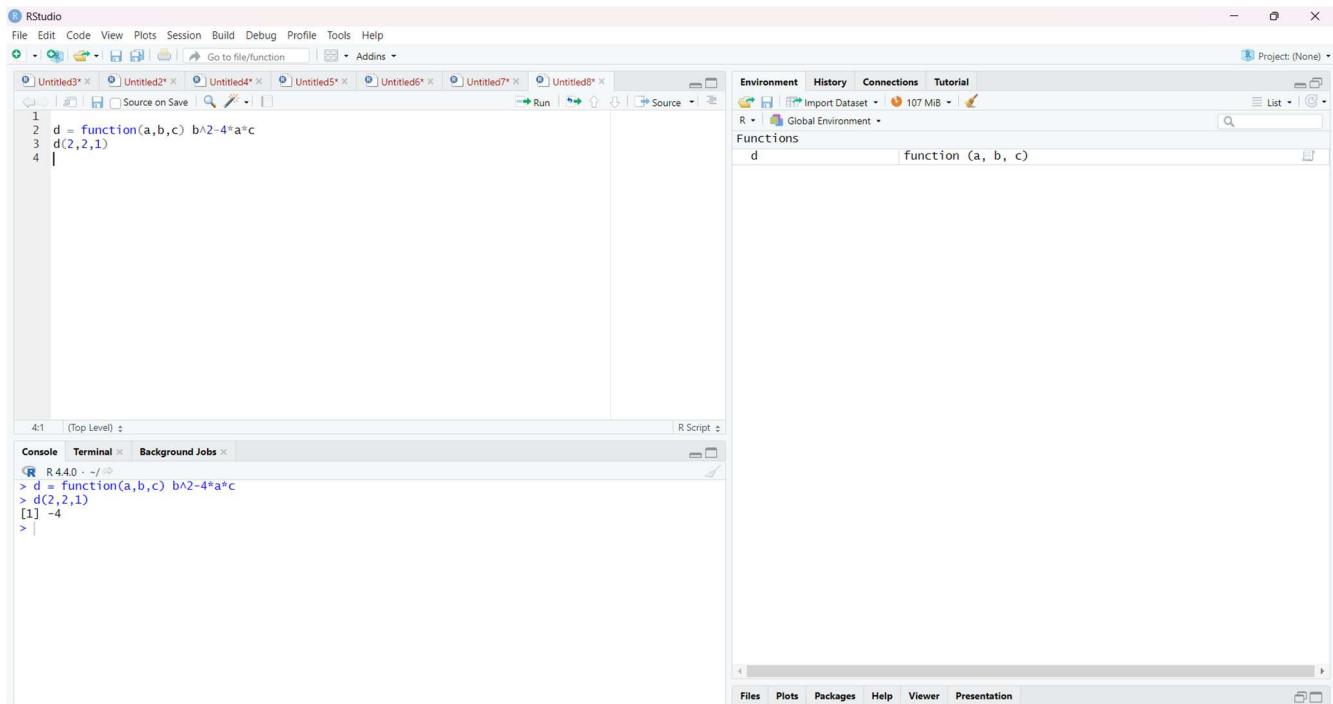
Definir Funciones y pasar Parámetros

Primero definimos que la variable **d** va a tener tres valores **a,b,c** que asignamos mediante **function**, después definimos la operación que se estará realizando con los parámetros de **d**, la cual es: $b^2 - 4 * a * c$ (b al cuadrado menos 4 por a y todo esto por c)

```
d = function(a,b,c) b^2-4*a*c
```

Ahora definimos los valores que tendrán los parámetros de **d** y a la vez la impresión del resultado

```
d(2,2,1)
```



Graficación de Funciones, utilizando la función **plot** la cual se utiliza para crear gráficos de manera general

```
g = function(x) sin(cos(x)*exp(-x/2))
```

```
plot(g, -8, -5, # se establece el rango del gráfico
```

```
lwd = 1, #Grosor de las líneas de la gráfica
```

```
main = "Gráfico de g", #Título del gráfico
```

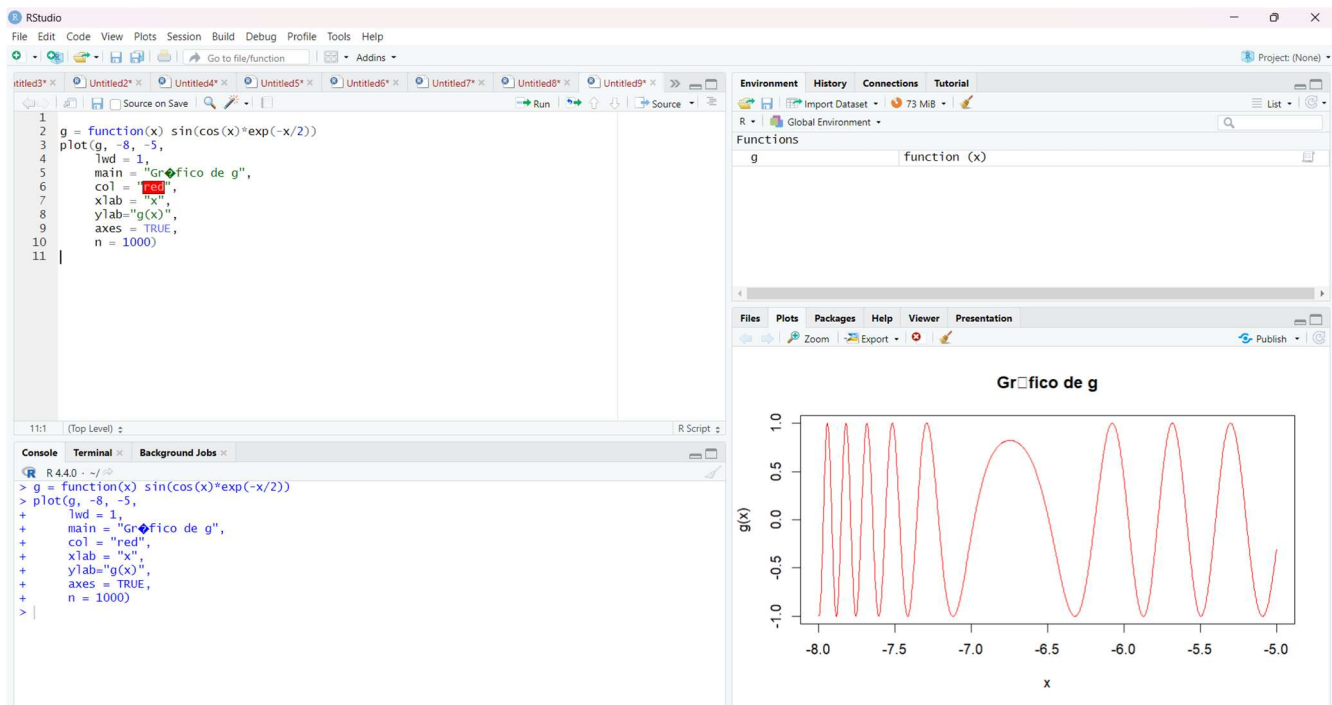
```
col = "red", # Color que tendrán las líneas de los gráficos
```

```
xlab = "x", # Etiquetas de x
```

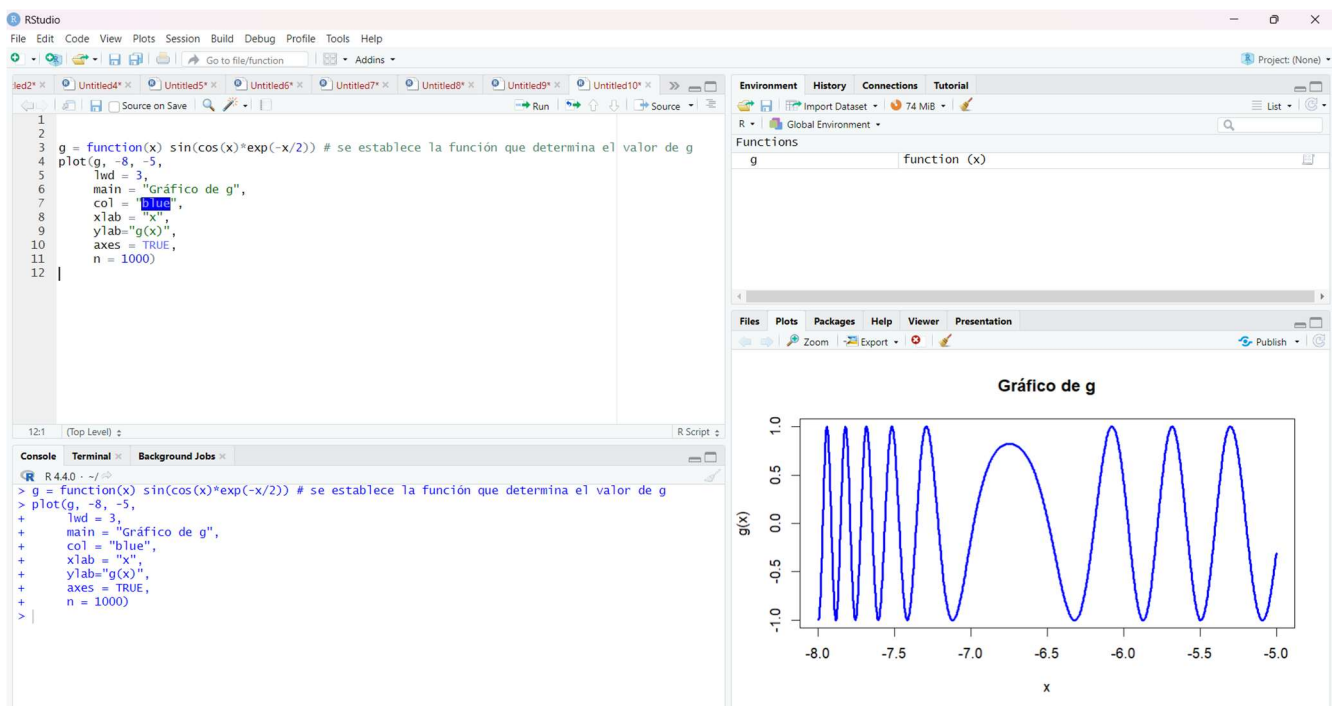
```
ylab="g(x)", # Etiquetas de y
```

axes = TRUE, # Ejes x,y visibles

n = 1000) # Números de puntos



Vamos a cambiar algunos datos para visualizar la diferencia:



CONCLUSIÓN

La importancia de los métodos numéricos radica en:

1. Instrumentos fuertes: Es una alternativa al método de solución analítica para la solución de ecuaciones lineales y no lineales.
2. Fácil acceso: debido a su constante desarrollo, hoy en día existe en el mercado una gran cantidad de software libre disponible para aplicaciones dentro de la Ingeniería; como el Rstudio, el cual se utilizó para el desarrollo de esta actividad.
3. Escalabilidad, en caso de no existir solución alguna, derivado del conocimiento de los métodos numéricos y programación van a permitir la generación de un nuevo programa que dé solución a la problemática que se esté presentando.
4. Adaptabilidad, debido a la gran cantidad de aplicación que tienen es posible aprovechar el potencial que tienen hoy en día las computadoras.

Se agrega dicha actividad a la plataforma de GitHub a través del siguiente link:

<https://github.com/22HADRIA/M-todos-Num-ricos>

REFERENCIAS ELÉCTRONICAS

Introducción a los Métodos Numéricos:

<http://itpn.mx/recursosisc/4semestre/metodosnumericos/Unidad%20I>

Algebra Lineal: Métodos Numéricos:

<https://www.ehu.eus/ebravo/contenidos/Algebra%20lineal/Numerico.pdf>

Métodos Numéricos:

<https://departamento.us.es/edan/php/asig/GRABIO/GBM/Tema4.pdf>

Método Numéricos para Ingenieros 5ta Edición

<http://artemisa.unicauca.edu.co/~cardila/Chapra.pdf>

<https://www.libreriaingeniero.com/2021/11/que-son-y-para-que-sirven-los-metodos-numericos.html>

Métodos Numéricos

http://cimogsys.esPOCH.edu.ec/direccion-publicaciones/public/docs/books/2023-04-20-144505-metodos_numericos.pdf