

CALCULADORA CIENTÍFICA

Objetivos y Motivación

Como estudiante de física que fui y amante de la programación que sigo siendo, quise hacerme una calculadora científica que me solucionara los problemas matemáticos más complejos en los primeros dos años de carrera.

Curiosamente, en primero de carrera, la programación se me daba bastante mal. Mi mentalidad siempre ha sido la misma, “metete en un proyecto en el que vayas a sudar pero que, cuando lo acabes, agradecerás una vez veas el resultado”. Así fue cómo decidí programar mi versión alternativa a “Wolfram Alpha”.

Resumen del Proyecto

Este proyecto va a simular una calculadora científica que interactúa con el usuario.

```
> calculadora()
[1] "Instale: library(matlib), library(Deriv), library(cubature), library(mosaicCalc), library(BB)"
[1] "Instale: library(Ryacas), library(mosaic) "
[1] "Indicaciones, responda todas las preguntas en minúsculas y sin tildes."
[1] "Para activar el comando, responda la palabra que está entre guiones: -palabra-"
[1] "-----"
[1] "-----"
[1] "Esta calculadora le permitirá hacer: "
[1] " un -sistema-(lineal), -ceros- de una funcion, -deri-var, -int-egrar, -ec.dif1-(primer orden),-e
c.dif2-"
[1] "-ec.dif3-(segundo orden no homogénea), -minimo- de una funcion, -auto-valores y autovectores, -si
st.nl- (no lineal)?:"
[1] "-limit-e de una función?"
¿Qué desea hacer?:
```

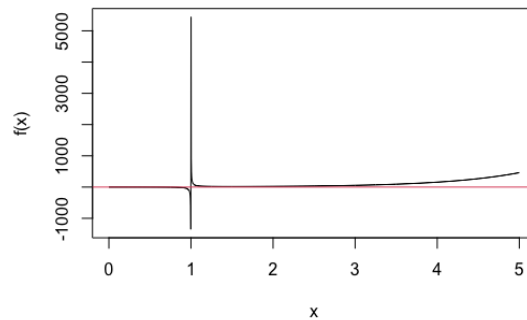
Me encargué de dar las indicaciones pertinentes como, por ejemplo, descargar las librerías necesarias o indicar los comandos para acceder a las herramientas de dicha calculadora.

Elijamos “ceros”, para encontrar dónde se anula una función. La secuencia de preguntas que deberemos seguir para obtener el resultado, es la siguiente.

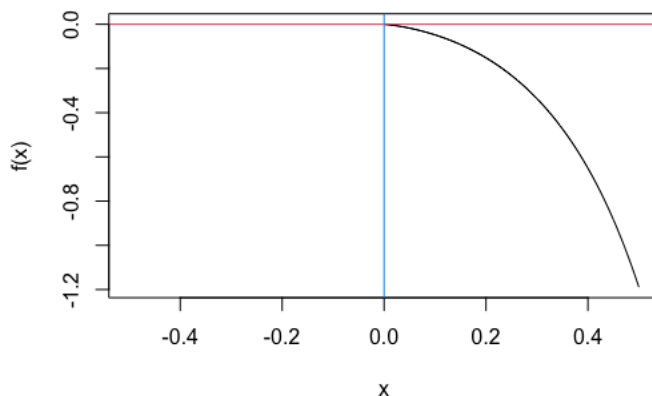
```

¿Qué desea hacer?:ceros
[1] "ahora cree una función tal que: f=function(x){...}"
[1] "y compile cero(f(x))"
[1] "-----"
> f=function(x){x*exp(x)*(1/log(x))}
> cero(f(x))
¿Quiere cambiar los límites?: si
¿Defina xmin?: -0.5
¿Defina xmax?: 0.5
Los ceros que se encuentran en el intervalo elegido son:
      x
1 0.0000
2 0.9999
¿Quiere volver a cambiar los límites?: si
¿Defina xmin?: -0.2
¿Defina xmax?: 0.2
Los ceros que se encuentran en el intervalo elegido son:
      x
1 0.0000
2 0.9999
¿Quiere volver a cambiar los límites?: no

```



Como podemos ver, nos pregunta si queremos ajustar el límite de los ejes para poder calcular los ceros de la función de una manera más precisa. Si la respuesta es afirmativa, volvemos a dictar los valores que consideremos y el programa re-dibujará la función en consecuencia.



Supongamos que estamos frente a una integral bastante compleja y la queremos resolver analíticamente.

```

¿Qué desea hacer?:int
¿Desea integrar de forma analitica o numerica?: analitica
[1] "Compile el comando: antiD( SU FORMULA ~ LA VARIABLE QUE DESEA INTEGRAR ) "
[1] "Ejemplo: antiD(x^2*y ~ x ) "
> antiD(x^2*y*(cos(pi*zx)+2) ~ x )
function (x, y, zx, C = 0)
(x^3 * y * cos(zx * pi) + 2 * x^3 * y)/3 + C
>

```

Se puede ver cómo en 3 líneas y con las indicaciones, obtenemos un resultado que podría sacarnos de dudas en caso de tenerlas.

Podríamos seguir probando pero la cuestión es ¿Quieres ver más pruebas o quieres probar tú?. Dejo a disposición mi código para que pueda, quizás, salvarte a ti también de algún apuro matemático.

