

A lo largo de esta guía les proponemos algunos ejercicios para que vayan resolviendo a medida que vayan viendo los videos. Estos mismos enunciados figuran en las TAREAS del documento pdf con la presentación de la clase; de todas formas, los recopilamos en este documento, por si facilita las cosas.

Tarea 1: Considerar la función

$$h(x) = \begin{cases} 0 & \text{si } x < 10 \\ \frac{x-10}{15-10} & \text{si } 10 \leq x \leq 15 \\ 1 & \text{si } x > 15 \end{cases}$$

1. Implementar una función **funcion.h** que dado un número devuelva el valor de la función h en ese número.
2. Graficar la función $h(x)$ para $x \in [5, 20]$. Exportá tu gráfico y subilo a nuestro documento compartido que encontrarás en acá.
3. Si quisiera cambiar las constantes 10 y 15 por parámetros $a < b$, respectivamente, de tal manera que puedan ser fijados por el usuario, ¿como debería reprogramarse la **funcion.h**?

Tarea 2: Combinemos bucles y condiciones.

1. Implementar una función **suma.positivos** que dado un vector v suma sus componente positivas; si no hay componentes positivas debe devolver el valor cero. Copiá el código en este documento compartido.
2. Calcular la suma de los números naturales positivos comprendidos entre los valores que le indicamos de manera personalizada en la hoja *suma positivos* que encontrarás acá y completar con el resultado obtenido en la columna D de la mencionada hoja de cálculo.
3. (*) Implementar una función **suma.si.hay.positivos** que dado un vector v suma sus componente positivas; si no hay componentes positivas debe devolver un cartel que diga "No hay positivos".

Tarea 3:

1. Crear una función **suma.cubilete** que simule arrojar el cubilete y dé por resultado la suma de las 5 caras obtenidas.
2. Invocar una vez la función **suma.cubilete** y completá la planilla aquí con el resultado obtenido.
3. Leemos la planilla y hacemos un histograma con los datos leídos.

4. Cada uno simula todo el proceso ahora con R: genera los 40 resultados del cubilete, calcula la suma y grafica un histograma con los resultados de su simulación.



5. Repetir en lugar de 40, 1000 veces el ítem anterior.

6. (*) Implementar una función **todos.seis** que cuenta la cantidad necesaria de lanzamientos hasta obtener el valor seis en todas la caras. ¿Cómo puede utilizar la función *suma cubilete* para verificar si sale seis en todas las caras?

