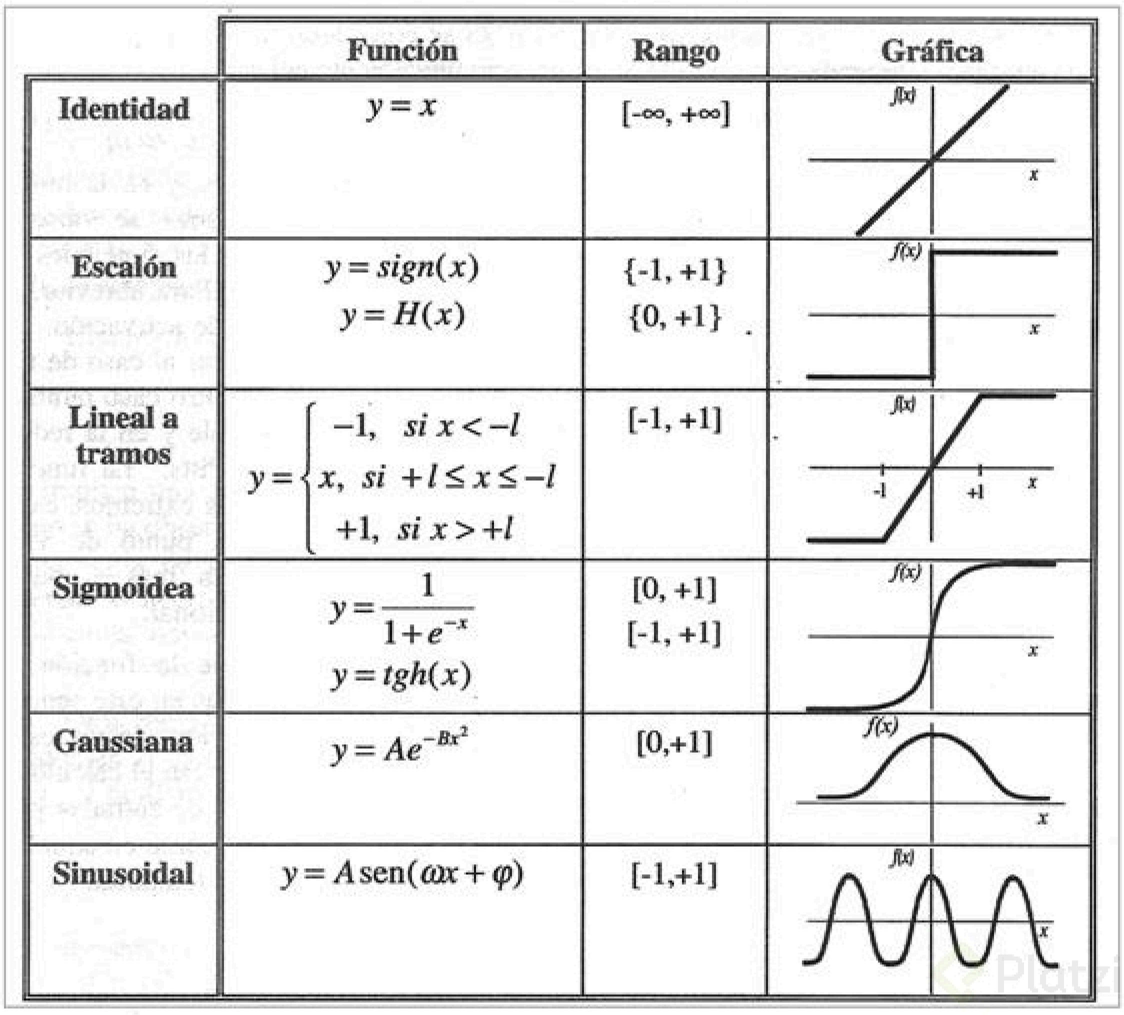
Funciones de activación

Existe en una parte del perceptrón y nos sirven para darle valores de salida diferente, a toda la combinación de entrada lineal del perceptrón.

Vamos a ver como se utilizan estas funciones de activación y como se implementan al final del perceptrón.



	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = sign(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Lineal a tramos	$y = \begin{cases} -1, & si\ x < -l \\ x, & si\ +l \leq x \leq -l \\ +1, & si\ x > +l \end{cases}$	$[-1, +1]$	
Sigmoidea	$y = \dfrac{1}{1+e^{-x}}$ $y = tgh(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0,+1]$	
Sinusoidal	$y = A\,\mathrm{sen}(\omega x + \varphi)$	$[-1,+1]$	

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt

#Configurando Latex
# Configuración de Matplotlib para usar LaTeX
plt.rcParams.update({
    "text.usetex": True,
    "font.family": "serif",
    "font.serif": ["Computer Modern Roman"],
    "text.latex.preamble": r"\usepackage{amsmath}"
})
```

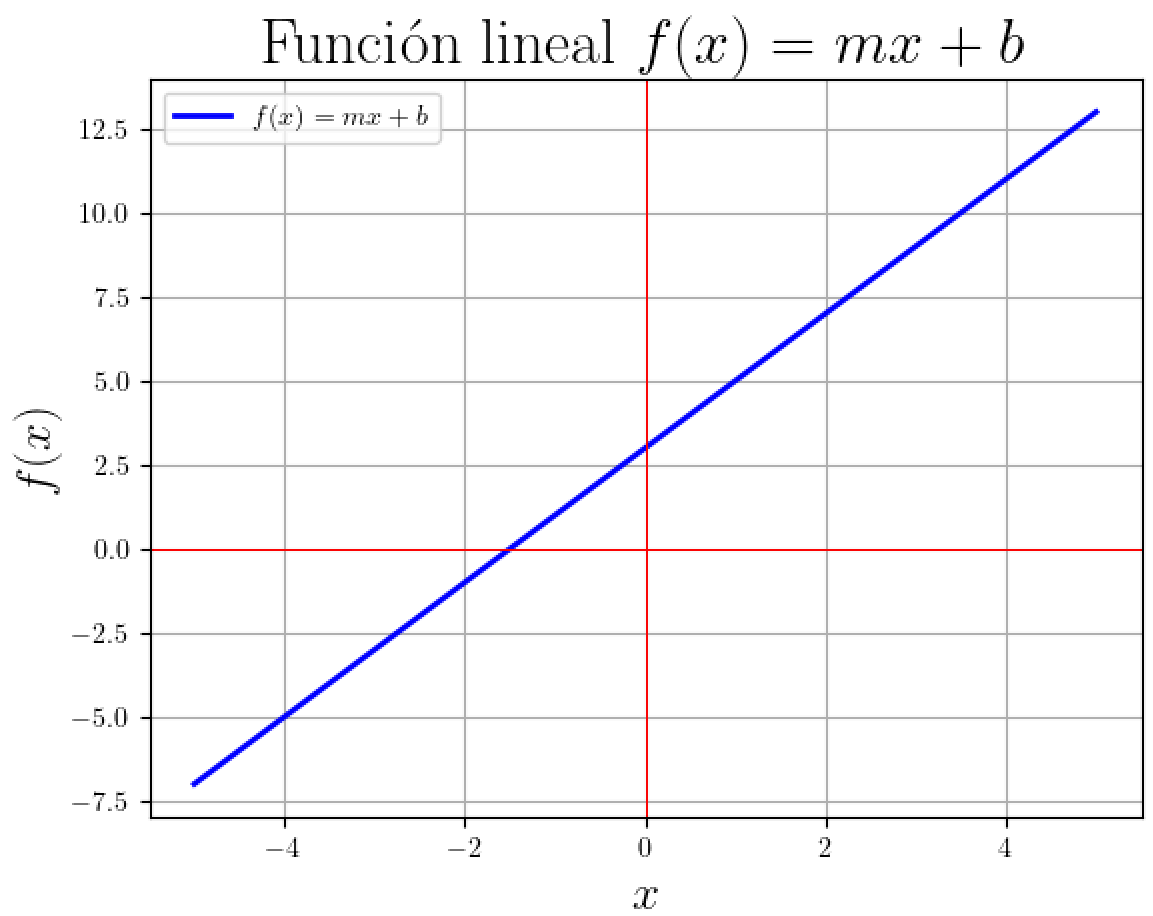
Función lineal

$$f(x) = mx + b$$

In [ ]:
```
# definiendo funcion parabola
def f(m,x,b):
  return m*x+b

#Definiendo variables
N = 1000
x = np.linspace(-5,5, num=N)

#Creando graficas
fig, ax = plt.subplots()
ax.plot(x,f(2,x,3),label=r'$f(x)=mx+b$',color='blue',linewidth=2)
ax.grid()
ax.axhline(y=0, color='r',linewidth=0.7)
ax.axvline(x=0, color='r',linewidth=0.7)
plt.title(r'Función lineal $f(x)=mx+b$',fontsize=23)
plt.xlabel(r'$x$',fontsize=18)
plt.ylabel(r'$f(x)$',fontsize=18)
plt.legend()
plt.show()
```



¿Para qué sirve?

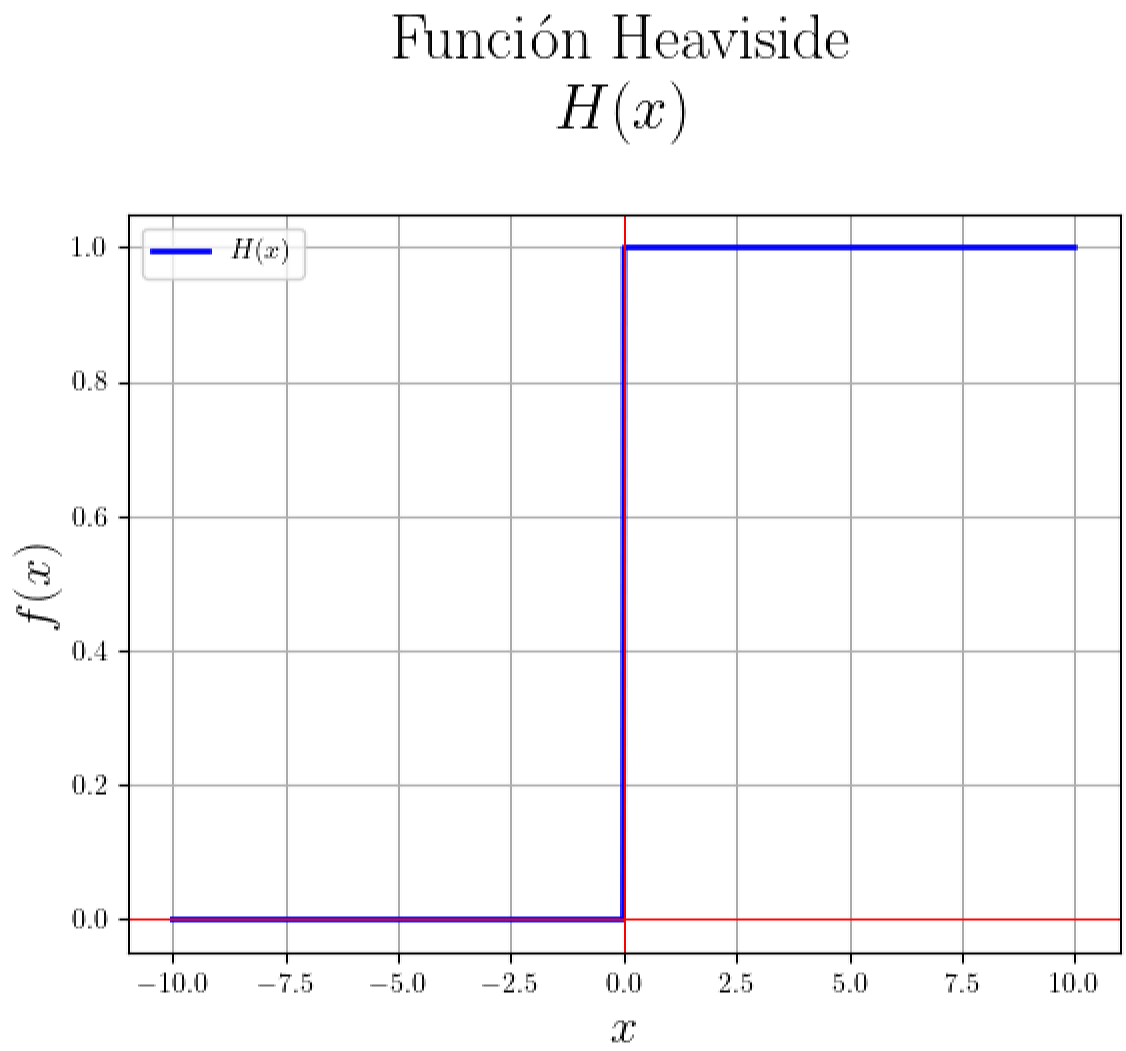
Todo lo que pasa mediante esta función regresa con el mismo valor.***¿Para qué sería bueno esto?*** Sirve para mantener valores durante un proceso, por ejemplo si tu quisieras mantener el valor de una venta que es a tráves del proceso de regresión lineal, pues lo que te interesa al final, es obtener la predicción de algún valor, es decir cuando sigues tendencias de algunos datos.

Función escalón o de Heaviside

$$H(x) = \begin{cases} 0, & \text{para, } x < 0 \\ 1, & \text{para. } x \geq 0 \end{cases}$$

In [ ]:
```
def H(x):
  Y = np.zeros(len(x))
  for idx,x in enumerate(x):
    if x>=0:
      Y[idx]=1
  return Y


N=1000
x = np.linspace(-10,10, num=N)
y = H(x)
fig, ax = plt.subplots()
plt.plot(x,y,label=r'$H(x)$',color='blue',linewidth=2)
plt.grid()
ax.axhline(y=0, color='r',linewidth=0.7)
ax.axvline(x=0, color='r',linewidth=0.7)
plt.title(f'Función Heaviside\n$H(x)$\n',fontsize=23)
plt.xlabel(r'$x$',fontsize=18)
plt.ylabel(r'$f(x)$',fontsize=18)
plt.legend()
plt.show()
```



Todos los valores que sean menores que cero, pues los va a volver 0 y pues todos los valores mayores a 1, los va a volver 1.

¿Para qué sirve?

Para poder **hacer clasificaciones categóricas**, esto nos podría indicar si algo está `[prendido,apagado],[existe, no existe]` es decir para jugar con valores binarios.
Esto en modelos de clasifican super sencillos es de gran utilidad.

Función Sigmoide

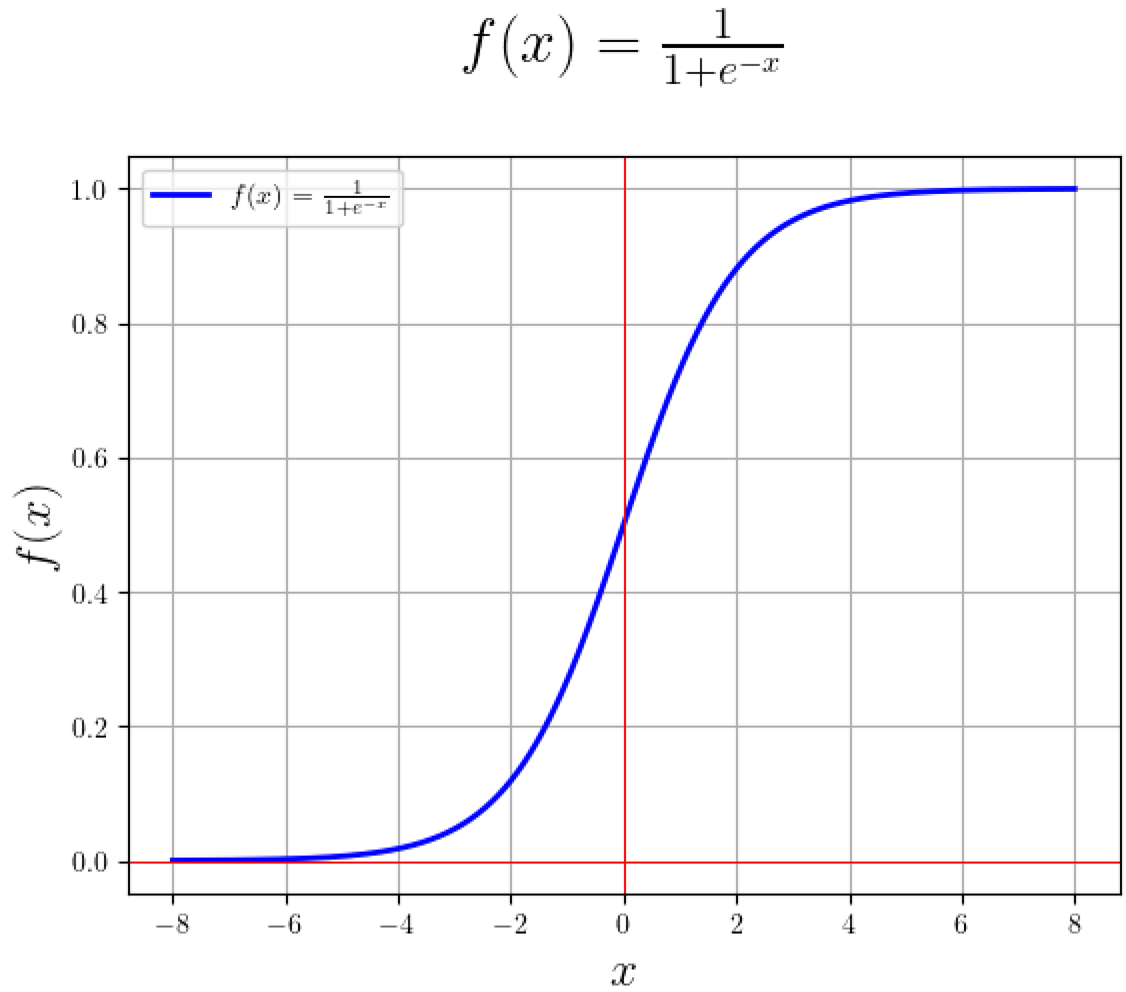
$$f(x) = \frac{1}{1+e^{-x}}$$

In [ ]:

```
def f(x):
  return 1/(1 + np.exp(-x))


N=1000
x = np.linspace(-8,8, num=N)
y = f(x)

fig, ax = plt.subplots()
plt.plot(x,y,label=r'$f(x)=\frac{1}{1+e^{-x}}$',color='blue',linewidth=2)
plt.grid()
ax.axhline(y=0, color='r',linewidth=0.7)
ax.axvline(x=0, color='r',linewidth=0.7)
plt.title(r'$f(x)=\frac{1}{1+e^{-x}}$'+'\n',fontsize=23)
plt.xlabel(r'$x$',fontsize=18)
plt.ylabel(r'$f(x)$',fontsize=18)
plt.legend()
plt.show()
```



¿Para qué sirve?

Para un ***modelo de regresión logistica*** y ¿por qué? sobre todo cuando hablamos de probabilidades.

> Porqué su **rango** va de $0-1$ y las probabilidades también van de *0 a 1*

IMPORTANTE
cuando el valor de $X=0$ el valor de $Y=0.5$

Función tangente hiperbólica

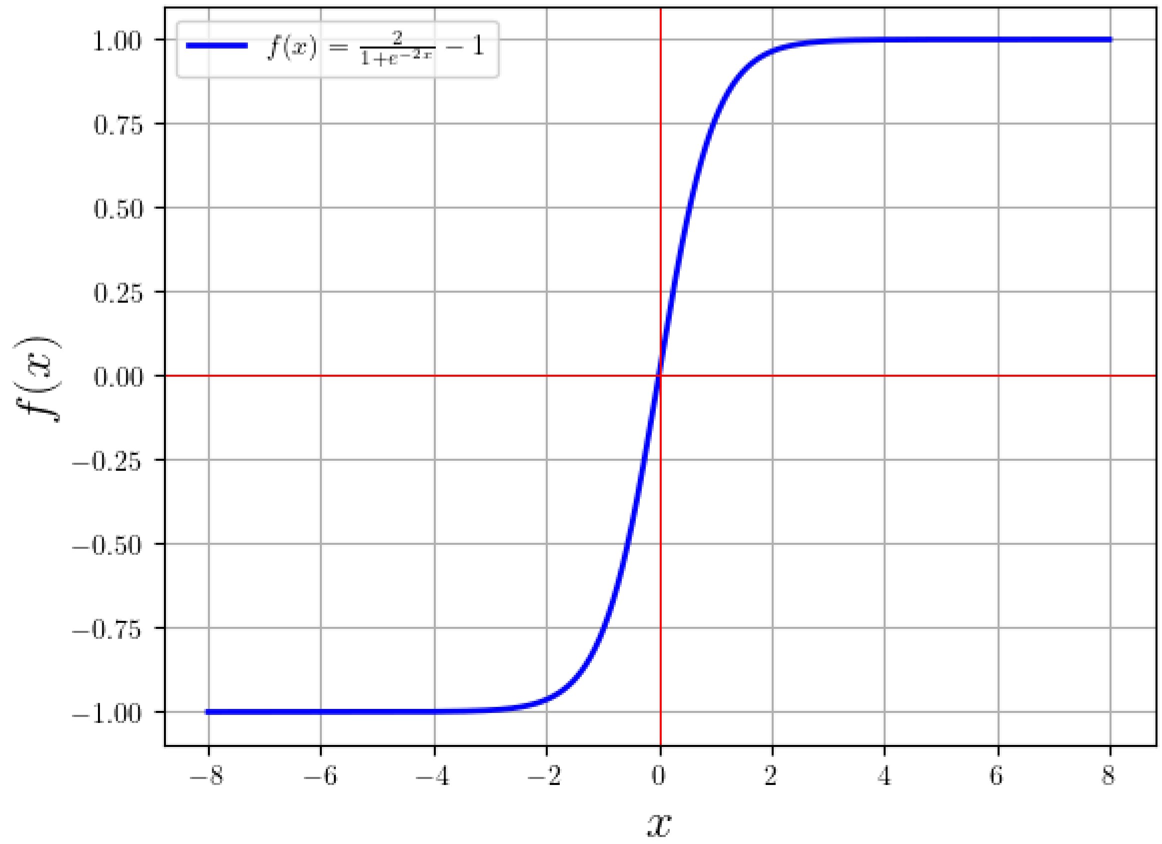
$$f(x) = \frac{2}{1+e^{-2x}} - 1$$

In [ ]:
```python
def f(x):
  return np.tanh(x)


N=1000
x = np.linspace(-8,8, num=N)
y = f(x)

fig, ax = plt.subplots()
plt.plot(x,y,label=r'$f(x)=\frac{2}{1+e^{-2x}}-1$',color='blue',linewidth=2)
plt.grid()
ax.axhline(y=0, color='r',linewidth=0.7)
ax.axvline(x=0, color='r',linewidth=0.7)
plt.title(r'$f(x)=\frac{2}{1+e^{-2x}}-1$'+'\n',fontsize=23)
plt.xlabel(r'$x$',fontsize=18)
plt.ylabel(r'$f(x)$',fontsize=18)
plt.legend()
plt.show()
```

$$f(x) = \frac{2}{1+e^{-2x}} - 1$$



¿Para qué sirve?

También se le conoce como una función de escalamiento y va a tener los valores de -1 a 1.

> Esta función tiene un problema con el algoritmo **bad propagation**. Siguen un patrón similar al `sigmoide`.

Función ReLU

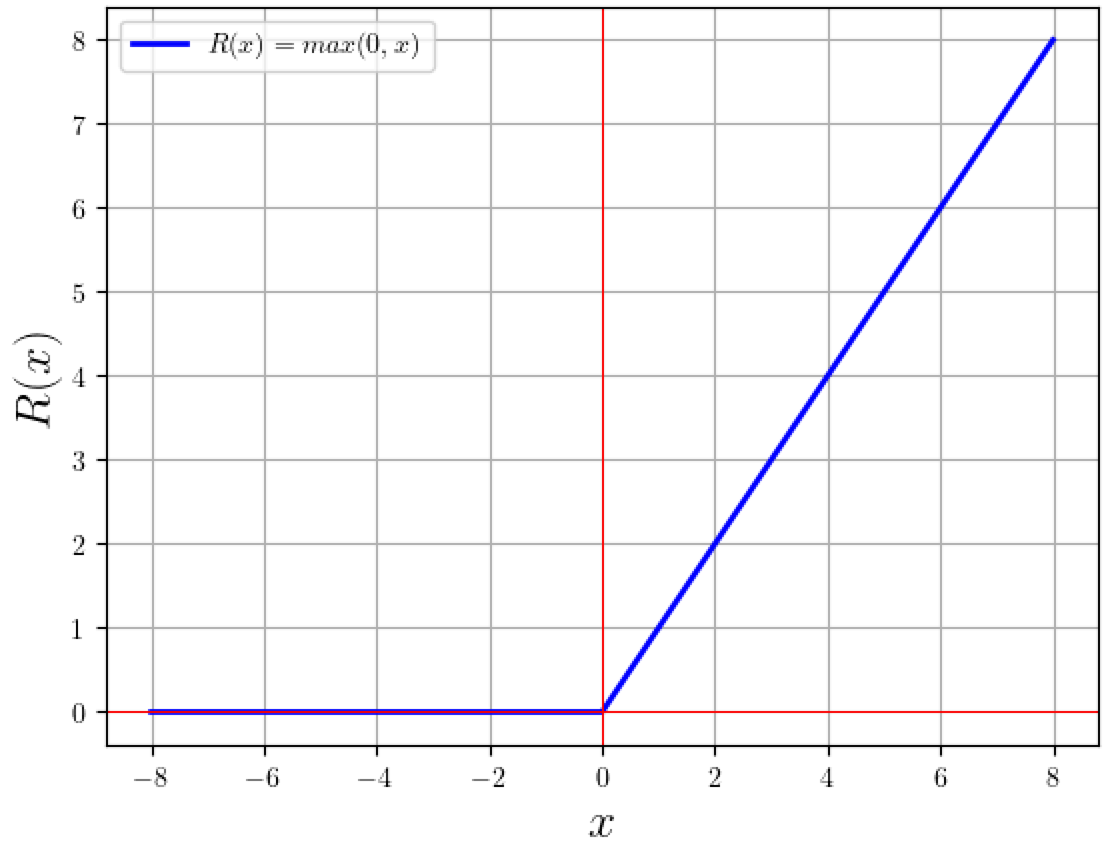
$$R(x) = max(0, x)$$

In [ ]:
```
def f(x):
  return np.maximum(x,0)


N=1000
x = np.linspace(-8,8, num=N)
y = f(x)

fig, ax = plt.subplots()
plt.plot(x,y,label=r'$R(x)=max(0,x)$',color='blue',linewidth=2)
plt.grid()
ax.axhline(y=0, color='r',linewidth=0.7)
ax.axvline(x=0, color='r',linewidth=0.7)
plt.title(r'$R(x)=max(0,x)$'+'\n',fontsize=23)
plt.xlabel(r'$x$',fontsize=18)
plt.ylabel(r'$R(x)$',fontsize=18)
plt.legend()
plt.show()
```

$$R(x) = max(0, x)$$



¿Para qué sirve?

Significa que si;

$$R(x) = \begin{cases} 0, & \text{para, } x < 0 \\ x, & \text{para. } x \geq 0 \end{cases}$$

En otras palabras para valores de $x < 0; R(x) = 0$ para valores de x menores a 0 la función vale 0, por otro lado para valores de $x > 0; R(x)$ para valores de x mayores a 0 la función vale $x, es decir va a empezar a existir un crecimiento.

> A esta función se le pone al final de un perceptrón, cuando nosotros queremos simular un proceso de neuronas muertas (cuando los valores tienden a ser 0 o negativos *es decir no existentes para el proceso que nosotros realizamos con nuestro modelo* y cuando tenemos un valor que si pueda procesar; es decir mayor a cero, entonces si vamos a tomar el valor a procesar).
>
> ### Digamos que la neurona va decir: ¿Existe? entonces lo tomo en cuenta, ¿no existe? entonces lo pongo como cero.
>
> Existe una variación de esta función que se llama Leaky RELU; lo que hace es que en los valores donde está el 0, los pone con una ligera pendiente, para que exista una pequeña ponderación.

Extra:

- Funciones de activacion
- Funciones de activacion
- Diferencias
- Post