

# Cálculo de una integra mediante el método Monte Carlo

## Código:

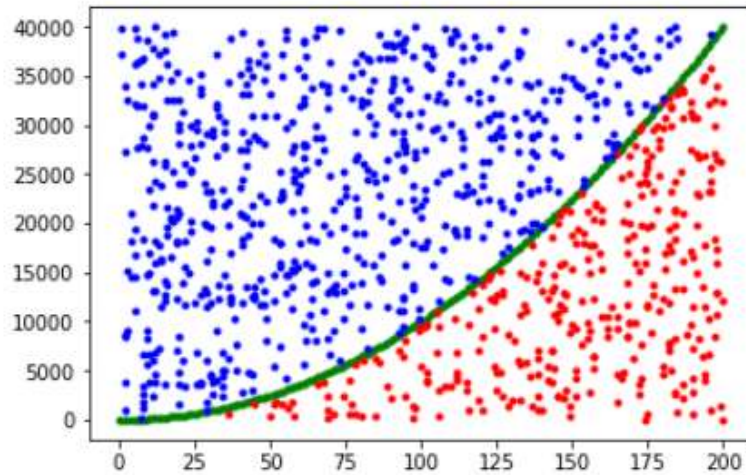
```
def integra_mc(fun, a, b, num_puntos = 10000):  
  
    i = 0  
    j = a  
  
    inferior = 0  
    maximo = 0  
  
    while(j <= b):  
        aux = fun(j)  
  
        if(aux > maximo):  
            maximo = aux  
            j = j + 1  
  
    tic = tm.time()  
  
    while(i < num_puntos):  
        rx = rand.randint(a,b)  
        ry = rand.randint(0,maximo)  
  
        if(ry < fun(rx)):  
            inferior = inferior + 1  
  
        i=i+1  
  
    res = (inferior/num_puntos)*(b-a)*maximo  
    toc=tm.time()  
  
    t_ejec = toc-tic  
  
    print("Tiempo de ejecucion: {} segundos".format(t_ejec))  
    return res
```

## Comentario:

Esta función escribe por pantalla el tiempo que le toma al algoritmo mediante generación de puntos aleatorios en la gráfica para posibles comparaciones. Devuelve un *float* con el valor de la integral en el intervalo (a,b)

### Gráfica resultante:

- En azul, puntos externos al área
- En rojo, puntos considerados para el cálculo de la integral



Se calcula el número de puntos rojos, se dividen entre todos los puntos generados, y se multiplica por el intervalo en el eje x y el máximo valor de la función en el intervalo

### Ejemplo de ejecución:

```
In [138]: prueba = integra_mc(funcion, 0, 200)
          print(prueba)
```

Tiempo de ejecucion: 0.0069866180419921875 segundos  
2848000.0