

En la presente práctica vamos a aprender a emplear la recursividad en la ejecución de funciones. Vamos a implementar el algoritmo de la función de Fibonacci, en su versión recursiva:

```
f_{0}=0  
f_{1}=1  
f_{n}=f_{n-1}+f_{n-2}
```

Vamos a implementar el algoritmo de la función de Factorial, en su versión recursiva:

```
function f(x){  
  if (n<=1) return 1;  
  else return n*f(n-1)  
}
```

Deberás crear un campo que recoja el número de iteraciones que quieres que recorra el algoritmo, con una pequeña validación de errores para que el número no sea menor que cero y que no se acepten caracteres distintos de dígitos.

El algoritmo deberá sacar por pantalla la sucesión calculada en el caso de Fibonacci, y el número resultante en el caso de Factorial, cuando se presione un botón.

Al mismo tiempo, deberemos averiguar el tiempo que tarda cada función en ejecutar el número de términos que le hayamos indicado.

Si esto te ha resultado fácil, te invito a insertar en el código HTML los valores de texto de la sucesión de Fibonacci o a emplear la función creada para calcular un número combinatorio. Te dejo debajo las pistas para su cálculo.

Los números combinatorios, coeficientes binomiales o combinaciones, se definen y denotan como:

$$C_m^n = \binom{m}{n} = \frac{m!}{n!(m-n)!}$$

en donde  $m$  y  $n$  son enteros y  $m \geq n > 0$ . El número combinatorio de arriba se lee como  $m$  sobre  $n$ .

.



May The Code be with you!!!