

## Ejercicio 1

El cálculo de números primos siempre ha sido un reto comunicacional, además, su cálculo es muy importe, ya que, existen múltiples aplicaciones con estos números.

Se pide al alumno que programe tres versiones diferentes de la función

**def GeneraPrimos (n:int) → list**

La función tendrá como parámetro el número de primos a generar y devolverá todos los primos generados en una lista. Las versiones serán:

1. Buscar si el número es primo recorriendo todos los posibles divisores (en ningún momento se sale del bucle en caso de que no lo sea).
2. Buscar si el número es primo recorriendo todos los posibles divisores (en el momento que no sea primo salir del bucle).
3. Un algoritmo que sea más eficiente que los anteriores.

Se pide al alumno medir tiempos de ejecución para generar 1000 primos y comparar resultados.

## Ejercicio 2

Se pide al alumno realizar un programa de un algoritmo de divide y vencerás que permita buscar un elemento en una lista ordenada de numeros aleatorios.

El prototipo de la funcion a hacer seria el siguiente (en caso de que el elemento no exista, se devuelve None y en caso de que # exista, que se devuelva el indice):

**def busqueda\_binaria(lista:list, lim\_l:int, lim\_h:int, valor:int) → int**

Para comprobar el rendimiento frente al algoritmo básico de busqueda, programa la siguiente funcion

**def busqueda\_simple(lista, val) → int**

que busque elemento a elemento y que devuelva el indice del elemento o -1 en caso contrario.

## Ejercicio 3

Se pide al alumno que programe una función que calcule el máximo de un vector de forma iterativa y recursiva y comparar resultados.