

L2 Practica 2. Medir tiempo

1.- Comencemos con el ejemplo clásico de racket para evaluar el tiempo que se toma ejecutar factorial de 5000. (O un valor parecido)

```
#lang racket/base

(time

(define (factorial n)

  (fact-iter 1 1 n))

(define (fact-iter product counter max-count)

  (if (> counter max-count)

      product

      (fact-iter (* counter product)

                  (+ counter 1)

                  max-count)))

( factorial 5000)

(void))
```

```
Welcome to DrRacket, version 7.9 [3m].
Language: racket/base, with debugging; memory limit: 256 MB.
cpu time: 31 real time: 45 gc time: 31

cpu time: 32 real time: 36 gc time: 32

cpu time: 32 real time: 35 gc time: 32

cpu time: 31 real time: 38 gc time: 31

cpu time: 32 real time: 41 gc time: 32
```

	Tiempo del CPU (Milisegundos)	Tiempo Real (Milisegundos)	Tiempo GC (Milisegundos)
Tiempo 1	31	45	31
Tiempo 2	32	36	32
Tiempo 3	32	35	32
Tiempo 4	31	38	31
Tiempo 5	32	41	32
Tiempo Promedio	31.6	39	31.6

2.- Usando el algoritmo de Fermat para hallar números primos de 12 cifras. Contarlos tomar el tiempo también

```
#lang racket/base

(time

(define (square x) (* x x))

(define (expmod base exp m)
  (cond ((= exp 0) 1)
        ((even? exp)
         (remainder (square (expmod base (/ exp 2) m))
                    m))
        (else
         (remainder (* base (expmod base (- exp 1) m))
                    m))))

(define (fermat n)
  (define (test a)
    (= (expmod a n n) a))
  (define (iter a)
    (if (< a n)
        (if (test a)
            (iter (+ a 1))
            null)
        n))
  (iter 1))

(define (range-help cur goal lst)
  (cond
    [(= cur goal) lst]
    [else (range-help (+ 1 cur) goal (append lst (list cur)))]))

(define (range num1 num)
  (range-help num1 num '()))

(filter (lambda (n) (not (null? n))) (map fermat (range 100000000 100001000))))
```

Welcome to [DrRacket](#), version 7.9 [3m].

Language: racket/base, with debugging; memory limit: 512 MB.

cpu time: 24660703 real time: 24889402 gc time: 37395

```
' (100000007  
  100000037  
  100000039  
  100000049  
  100000073  
  100000081  
  100000123  
  100000127  
  100000193  
  100000213  
  100000217  
  100000223
```

```
100000231  
100000237  
100000259  
100000267  
100000279  
100000357  
100000379  
100000393  
100000399  
100000421  
100000429  
100000463  
100000469  
100000471  
100000493
```

```
100000541  
100000543  
100000561  
100000567  
100000577  
100000609  
100000627  
100000643  
100000651  
100000661  
100000669  
100000673  
100000687  
100000717  
100000721
```

100000721
100000793
100000799
100000801
100000837
100000841
100000853
100000891
100000921
100000937
100000939
100000963
100000969)

Nota: Mi programa permite evaluar un numero de 12 cifras pero tarda días en hacerlo y mi laptop no está en condiciones de hacerlo.