

## Ejercicio Grupal de JS: Iterations.

### PREGUNTAS

Usamos ES6/Node. Ejercitamos patrones de iteración (control) usando generadores. Usamos los conceptos de Iterator y adicionalmente “memoización”.

Estudie el modelo de código adjunto. Revisar los casos de uso en `test_iterations` y `test_fib`. Estos archivos no deben cambiarse.

1. Usando `iterations.iterator` implemente según se pide en `test_iterations` en el caso tres (ver y trabajar en `question1.js`). Se requiere un generador que dada una lista `list` y un máximo `max` genere enteros no negativos, que no sean divisibles por ningún elemento en `list` y que sean menores o iguales que `max`.

Salida esperada:

```
ITERATIONS:node test_iterations.js
>>> Test Iterations <<<
*** Generate (testing iterator) ***
3
6
9
12
15
18
21
24
27
30
*** Iterate (Testing Iterable Class) ***
3
6
9
12
15
18
21
24
27
30
*** Iterate (Q1 generator) ***
list = [2,3,5,7] max = 50
v = 1
v = 11
v = 13
v = 17
v = 19
v = 23
v = 29
v = 31
v = 37
v = 41
v = 43
v = 47
```

2. Reimplemente `fibo_iter` usando estilo FP y generadores (y features DRY apropiados de ES6) de forma que funcione `test_fib` como antes del cambio

Salida esperada luego de re-implementar `fibo_iter`

```
ITERATIONS:node test_fibo.js 25 50
>>> Testing fibo iterator : n = 25 max = 50
2) 2
3) 3
4) 5
5) 8
6) 13
7) 21
8) 34
9) 55
10) 89
11) 144
12) 233
13) 377
14) 610
15) 987
16) 1597
17) 2584
18) 4181
19) 6765
20) 10946
21) 17711
22) 28657
23) 46368
24) 75025
25) 121393
```