

# Pràctica 1: Programes concurrents amb Erlang

Programació Concurrent i en Temps Real — iTIC

Antoni Escobet Canal

23 d'agost de 2022

## Índex

<b>1 Organització</b>	<b>1</b>
1.1 Lliurament . . . . .	1
<b>2 Exercicis</b>	<b>1</b>

## 1 Organització

Aquesta sessió s'organitza com una seqüència de problemes de dificultat creixent l'objectiu dels quals és implementar petits programes escrits en Erlang. Amb l'objectiu de reforçar l'hàbit d'usar sistemes de control de versions, cal desenvolupar la pràctica amb el suport del sistema que ofereix <http://escriny.epsem.upc.edu>.

### 1.1 Lliurament

Cal lliurar els diferents fitxers font dels exercicis a través d'Atenea en la data fixada. Cal que el desenvolupament es faci usant Subversion a través de les facilitats que ofereix <http://escriny.epsem.upc.edu>.

## 2 Exercicis

EXERCICI 2.1 Escriviu un mòdul Erlang el que exporti les següents funcions:

- Una funció amb signatura `interseccioM(L1, L2, M)` que retorna la llista d'elements que pertanyen a les dues llistes simultàniament L1 i L2 i que son divisibles per M.

```
1> ex1:interseccio([a,b,a,12,c,6,30], [a,x,12,6],2).  
[6,12]
```

- Una funció amb signatura `filtreM(L,M)` que donada una llista retorna una nova llista amb tots els valors numèrics majors que M.

```
1> ex1:filtreM([1,2,7,a,b,5],4).  
[7,5]
```

- Una funció amb signatura `ordenar(L)` que donada una llista de sencers retorni la llista ordenada de major a menor. Per exemple:

```
1> ex1:ordenar([4,7,1,12,2,4]).  
[12,7,4,4,2,1]
```

Aquesta funció s'ha de fer amb l'algoritme d'ordenació GrupX.

Algoritme d'ordenació:

Grup 1 : Insertion

Grup 2 : Selection

Grup 3 : Bubble

Grup 4 : Shell

Grup 5 : Merge

Grup 6 : Heap

Grup 7 : Quick

Grup 8 : Quick3

Podeu consultar l'algoritma d'ordenació que us toca a la pàgina web: <https://www.toptal.com/developers/sorting-algorithms>

Implementeu les tres funcions (les que pugueu) de tres formes diferents:

- Recurssiu
- Recursiu amb acumuladors
- llistes de comprensió.