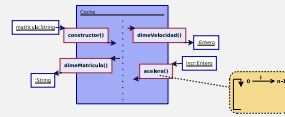


Grado Tecnologías Interactivas

Programación 2



Práctica 2

UNIVERSIDAD
POLITECNICA
DE VALENCIA

Escola Politècnica Superior de Gandia

DSIC

Departament de Sistemes Informàtics i Computació

Práctica 2

¡ Atención !

- ▷ Se recuerda que las prácticas deben prepararse antes de acudir al aula informática, anotando en el enunciado las dudas que se tengan.
- ▷ Los diseños y algoritmos que se piden en esta práctica deben escribirse en la libreta de apuntes para poder ser revisados.
- ▷ Utiliza *git* en cada ejercicio haciendo *commit* cada vez que consigas un "hito".
- ▷ La realización de las prácticas es un trabajo individual y original. En caso de plagio se excluirá al alumno de la asignatura. Por tanto, es preferible presentar el trabajo realizado por uno mismo aunque éste tenga errores.



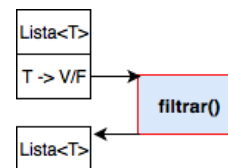
1

Filtros, Mapas, Reductores

1.1

Filtros

El diseño de una función que filtra una lista de elementos (sin importar su tipo) utilizando una función "condición" es el siguiente:

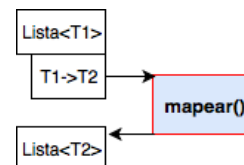


Implementa la función `filtrar()`. Pruébala de forma automática seleccionando de una lista de palabras, aquellas que contengan la letra 'x'. (Para facilitar la programación, hay que utilizar siempre que sea posible funciones de biblioteca, como por ejemplo el método `includes()` de los string).

1.2

Mapas

El siguiente es el diseño de una función que aplica una transformación a cada elemento de una lista para obtener una nueva lista.



Implementa la función `mapear()`. Pruébala de forma automática transformando una lista de palabras en una lista con las longitudes de cada palabra de la primera lista.



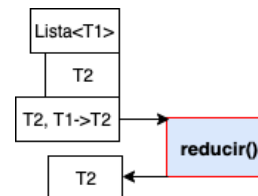
1.3

Reductores

Un reductor "colapsa" una lista de elementos en sólo uno, aplicando una función de forma repetida sobre los mismos. Por ejemplo, podemos reducir 1,2,3,4,5 a su suma 15 utilizando la operación + y un primer valor 0 de esta forma:

$$0 + 1 \rightarrow 1$$
$$1 + 2 \rightarrow 3$$
$$3 + 4 \rightarrow 7$$
$$7 + 4 \rightarrow 11$$
$$11 + 5 \rightarrow 15$$

Veamos el diseño de la función `reducir()`.



Implementa la función `reducir()`. Pruébala de forma automática con el anterior ejemplo.



2

All together now

Utilizando las anteriores funciones `filtrar()`, `mapear()` y `reducir()`; escribe trozos de código que, a partir de esta lista de palabras

```
var palabras = ["Es", "ahora", "tu", "oportunidad", "para", "aprovechar", "este", "día"]
```

1. obtenga una nueva lista con cada palabra escrita del revés
2. calcule el total de caracteres de la frase completa
3. calcule cuántas palabras tienen más de 5 letras
4. obtenga una palabra que sea la concatenación de aquéllas con 3 letras o menos.



28 febrero 2020