

# WUOLAH



cliptoner

[www.wuolah.com/student/cliptoner](http://www.wuolah.com/student/cliptoner)



20597

## Febrero2016-Ejercicio3.pdf

? Exámenes RESUELTOS | ADDA



2º Análisis y Diseño de Datos y Algoritmos



Grado en Ingeniería Informática - Tecnologías Informáticas



Escuela Técnica Superior de Ingeniería Informática  
US - Universidad de Sevilla

**Coucke's Academy**  
BY SARAH COUCKE, TEACHING SINCE 2005  
[www.couckesacademy.es](http://www.couckesacademy.es)

**BRITISH COUNCIL** **Aptis** **Cambridge English** **TRINITY**  
Exam Preparation Centre

Get your English certificate now!

### Nervión

Avenida San Francisco Javier 24,  
Planta Baja, Módulo 12C  
954 65 98 99 - 605 54 50 19  
[nervion@couckesacademy.es](mailto:nervion@couckesacademy.es)

### Macarena

Calle Don Fadrique 19  
954 38 51 02 - 636 64 90 58  
[macarena@couckesacademy.es](mailto:macarena@couckesacademy.es)

**Ejercicio 3: DyV**

Se dispone de una lista de canciones mp3 almacenadas en la nube, donde se conoce el nombre y la duración de cada canción. El sistema mantiene las canciones en una lista ordenada por título y se requiere implementar un algoritmo que devuelva la canción (o lista de canciones en caso de repetición) que ocuparía una determinada posición según la duración de las canciones (primero las canciones más cortas). Dicho algoritmo no debe ordenar toda la lista por completo para evitar la complejidad que ello implica en el caso medio.

Considere que existe una clase Canción que tiene las propiedades nombre (String) y duración (int), siendo su orden natural la ordenación por la duración.

**Ejemplo:** Para la lista siguiente, en la lista ordenada por la duración de las canciones, la posición 0 sería ocupada por la canción 3, y la 6 por la canción 4.

Canción 1	Canción 2	Canción 3	Canción 4	Canción 5	Canción 6	Canción 7
3:43	3:26	2:50	6:26	2:58	4:52	3:20

**Se pide:**

1. Rellene la ficha propuesta (partes marcadas como //TODO).
2. Implemente la función `List<Cancion> getCancionOrdDur(List<Cancion> d, int pos)` en lenguaje Java. Recuerde que la duración de las canciones pueden repetirse, en cuyo caso se devolvería la **lista de canciones** que tienen dicha duración.

**Puede hacer uso de las funciones:**

- `Pair<Integer,Integer> bh (List<T> l, T e, int i, int j)`, de complejidad lineal con respecto a  $j-i$ , y que reordena la lista  $l$  según el algoritmo de bandera holandesa. Devuelve una tupla con la posición en la que empiezan los elementos iguales a  $e$  y la posición en la que empiezan los elementos mayores que  $e$ .
- `T ep(List<T> l, int i, int j)`, de complejidad constante que devuelve un elemento aleatorio de la lista  $l$  en el intervalo  $(i, j]$ .

**IMPORTANTE:** Se prohíbe usar en la solución cualquier estructura de datos auxiliar ordenada por la duración de las canciones.



ADDA/EDA

Examen Febrero

Curso 2015/2016

Ficha Examen	
Canciones	
<b>Técnica:</b> Divide y Vencerás sin Memoria	
<b>Propiedades Compartidas:</b>	$d: List<Cancion>$ $pos, int$
<b>Propiedades Individuales:</b>	$// TODO$ $i, int$ $j, int$
<b>Solución:</b> Lista de canciones que ocuparían la posición pos de la lista ordenada por duración de las canciones (si las duraciones no se repiten la lista sólo podrá incluir una canción)	
<b>Inicialización</b> $getCancionOrdDur(d, pos) = getCancionAux(0, d.size()) // TODO$	
<b>Generalización:</b> $// TODO$ $getCancionAux(i, j) = \begin{cases} d.subList(a, b) & (pos \geq a \text{ y } pos < b) \\ getCancionAux(i, a) & pos < a \\ getCancionAux(b, j) & pos \geq b \end{cases}$ $p = ep(d, i, j)$ $<a, b> = bh(d, p, i, j)$	

```

private <T extends Comparable<T>> List<T> getCancionAux(List<T>
d, int pos, int i, int j) {
    List<T> res;
    T e = ep(d, i, j);
    Pair<Integer, Integer> ab = bh(d, e, i, j);
    if(pos >= ab.a && pos < ab.b){
        res = new ArrayList<T>();
        res.addAll(d.subList(ab.a, ab.b));
    }else{
        if(pos < ab.a){
            res = getCancionAux(d, pos, i, ab.a);
        }else{
            res = getCancionAux(d, pos, ab.b, j);
        }
    }
    return res;
}

```

