

# ***Proyecto EDI 18-19***

## ***Tercera Entrega***

Estudiante: Óscar Flores Gil  
DNI: 80227040G  
Grupo de prácticas: Lunes 17:30  
Profesor de prácticas: M<sup>a</sup> Angeles Mariscal

Estudiante: Rubén Costa Barriga  
DNI: 76042246Z  
Grupo de prácticas: Lunes 17:30  
Profesor de prácticas: M<sup>a</sup> Angeles Mariscal

Fecha de entrega: 09/06/2019  
Convocatoria: Ordinaria

Descripción del problema.....	3
Solución.....	3
Alternativas de diseño.....	3
Evaluación del diseño.....	3
Conclusiones.....	3
Conclusiones acerca del proyecto.....	3
Opinión personal y conclusiones.....	3

## Descripción del problema

El problema que se nos presenta es la programación de un archivo de tal forma que, por medio de cuatro algoritmos, ejecute operaciones que:

1. Genere un fichero con todos los árboles de una vía que se pedirá al usuario.
2. Genere un fichero en el cual cada línea aparezca el nombre de la vía y el número de árboles de una especie dada por el usuario.
3. Genere una estructura de árbol con todas las vías de un género dado por el usuario. Al devolver el árbol, por consola se muestra la información indicando el barrio donde se ubica cada vía.
4. Escriba en un fichero toda vía que comience por una subcadena específica, que contenga uno más barrios con esa misma subcadena y el número de árboles. Solo deben aparecer las vías que contengan árboles.

## Solución

La solución realizada para cada algoritmo:

1. Mediante un bucle recorreremos el conjunto de vías en árbol hasta encontrar la misma vía introducida con un comparador. Una vez encontrada, recorreremos la lista de árboles, generando un fichero con todos los árboles.  
La complejidad es  $O(\log n)$ , en vez de  $O(n)$  en la parte de la búsqueda de las vías, ya que se usa la estructura de árbol.
2. Mediante una cadena dada, buscamos dentro de cada barrio y a su vez, en cada vía, los árboles con esa misma especie. Una vez dentro del conjunto de árboles, gracias a un contador, vemos cuantos árboles de esa especie hay, generamos un fichero y escribimos en él el nombre de la vía y el número dado por el contador.
3. Mediante una cadena dada, insertamos de forma recursiva en árbol todas las vías que sean del género de la cadena. Luego llamamos a un módulo que recorre el árbol para mostrarlo por pantalla.
4. Creamos una lista nueva, en la que pondremos todos los elementos pedidos. Posteriormente comprobamos todos los elementos y si coinciden, comprobamos si tiene un árbol, si es así lo añadimos a la lista.

## Alternativas de diseño

Podríamos usar solo listas o pilas y colas, pero usar los árboles facilita la labor de programar y la complejidad es mucho menor y la unión de las dos nos ha facilitado el trabajo.

## Evaluación del diseño

Se debe completar la siguiente tabla donde se detalle la complejidad de los algoritmos implementados en el proyecto. Para ello, se deben comparar su complejidad en el mejor, peor y caso medio.

**Tabla 1: Complejidad algoritmos**

	Mejor Caso	Caso Medio	Peor caso
--	------------	------------	-----------

Algoritmo 1	$O(n * \log n)$	$O(n^2 * \log n)$	$O(\log n * n^2)$
Algoritmo 2	$O(n)$	$O(\log n)$	$O(\log n)$
Algoritmo 3	$O(\log n)$	$O(n * \log n)$	$O(n^2 * \log n)$
Algoritmo 4	$O(n)$	$O(n^2)$	$O(n^2)$

## Conclusiones

### ***Conclusiones acerca del proyecto***

Completado ya el proyecto podemos ver el uso de las diferentes estructuras de datos, concluyendo que la estructura de árbol es la más útil a la hora de organizar una lista de elementos. En ellas podemos remarcar que el uso de listas es mas intuitivo que el uso de árboles pero a su vez los árboles son mas cómodos a la hora de trabajar.

También hemos podido observar la interacción entre las clases, su utilidad y como interactua una con otras.

### ***Opinión personal y conclusiones***

Empezamos por la clase de árbol y así avanzamos por cada clase respectivamente. Al principio el uso de los árboles y su recursividad era un tanto lioso pero una vez comprendido el funcionamiento de ellos en tutorías su uso ha sido bastante útil.

Todo esto junto al las demás clases ya aprendidas de las anteriores entregas hemos podido unir todo en un proyecto de una manera fluida.

Cuesta hacerse a la idea de como funcionan las clases pero una vez captado el concepto y aprendido, podemos usarlas con soltura y unir las entre ellas.