

Pablo Hernandez Martinez, [pablo.hernandez.martinez@udc.es](mailto:pablo.hernandez.martinez@udc.es)

Marcelo Ferreira Sánchez, [marcelo.fsanchez@udc.es](mailto:marcelo.fsanchez@udc.es)

## **Fases de desarrollo**

Lo primero que hicimos fue crear los dos archivos: la clase Activity el archivo principal: main. La clase Activity tiene los atributos que utilizaremos más adelante (nombre, número de participantes, precio y duración). Además, importamos todos los archivos necesarios para cubrir las dependencias del programa, donde importamos la clase AVL. Esta clase define de forma abstracta un árbol de búsqueda binaria con sus respectivos métodos heredados. Esta estructura de datos nos permitirá manejar de forma eficiente las distintas actividades. Una vez que los getters, setters y demás métodos (sobreescribimos el método mágico para el operador <) estaban implementados para la clase Activity, empezamos a trabajar en el archivo main. Primero nos aseguramos de que los datos se leyesen e importasen correctamente a partir del archivo .txt provisto. Tuvimos que cambiar la codificación por defecto de la función open() ya que no era capaz de leer bien las tildes en un principio. A continuación, procedimos a escribir el código para añadir correctamente (y de manera ordenada) los objetos Activity a los árboles AVL. Decidimos ordenarlos alfabéticamente. Después de esto programamos cada una de las funciones en main que el programa debía ser capaz de ejecutar. La primera es realizar la suma de actividades, que devuelve un árbol con todas las actividades (sin repetir) que se encuentran en alguno de los árboles. La segunda hace lo mismo, pero solo conserva las actividades compartidas por ambos árboles. También importamos una librería (time) para medir el tiempo de ejecución del programa e intentar mejorarlo a lo largo del desarrollo. Por último, nos ocupamos de documentar correctamente todas las clases, métodos y funciones, y redactamos este documento. Por supuesto a lo largo de todo el desarrollo nos encontramos con multitud de bugs, errores y fallos en la lógica que tuvimos que solventar para que el programa funcionase.